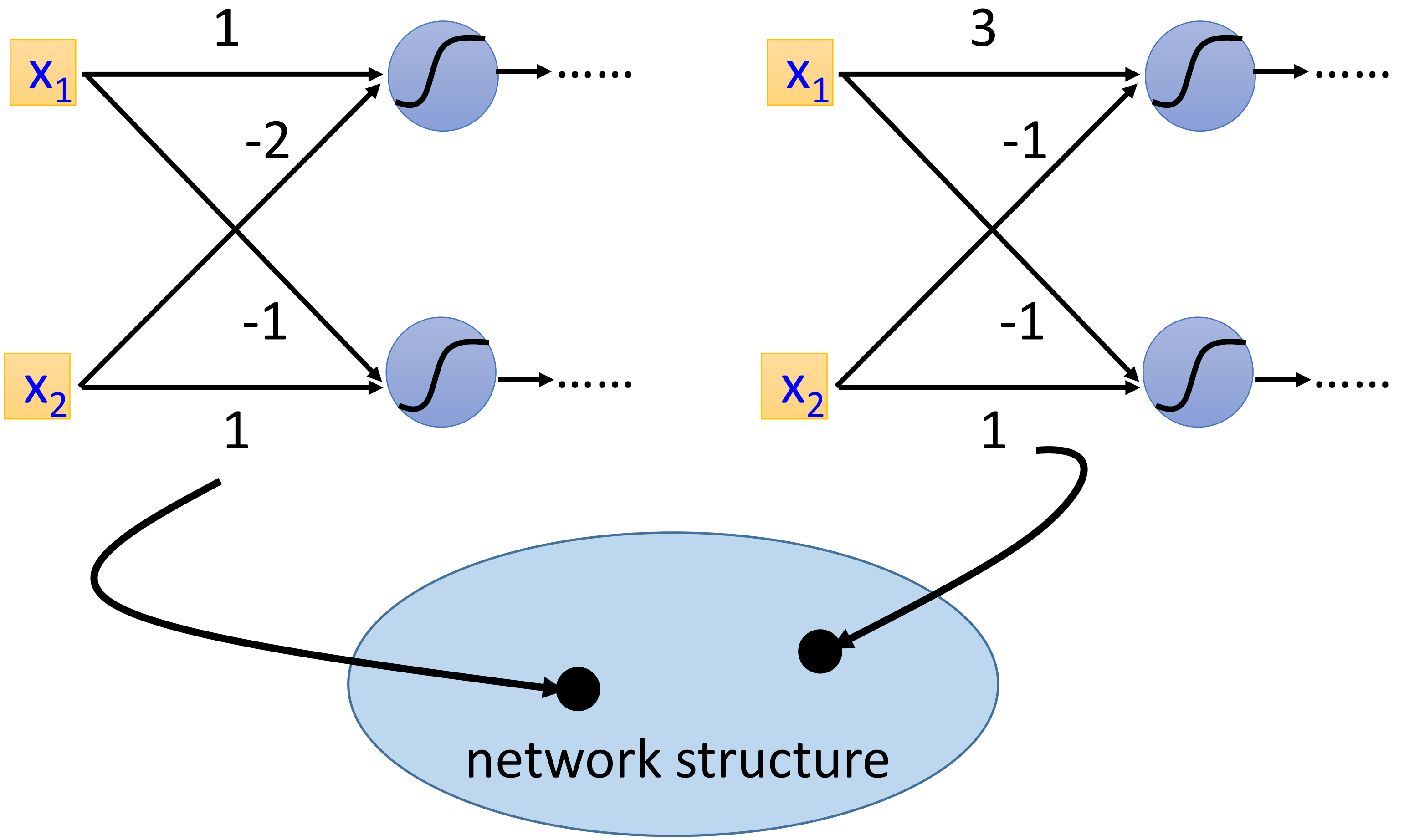
Theory I: Why Deep Structure?

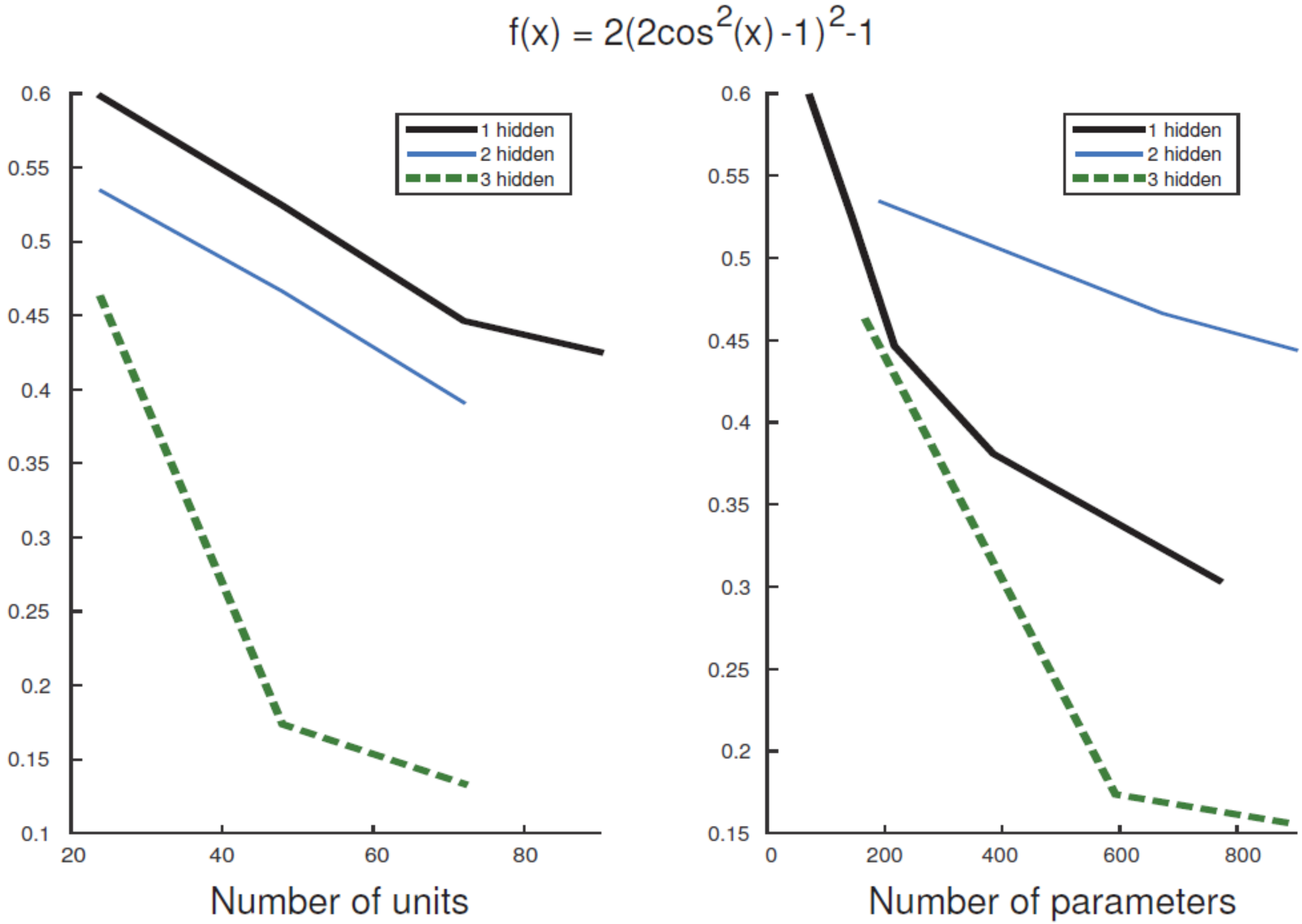
李宏毅
Hung-yi Lee

Review



Given structure, each set of parameter is a function.

The network structure defines a function set.



Source of image: 在比較的時候希望在參數量相同的情況下調整network架構

https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewPaper/14849

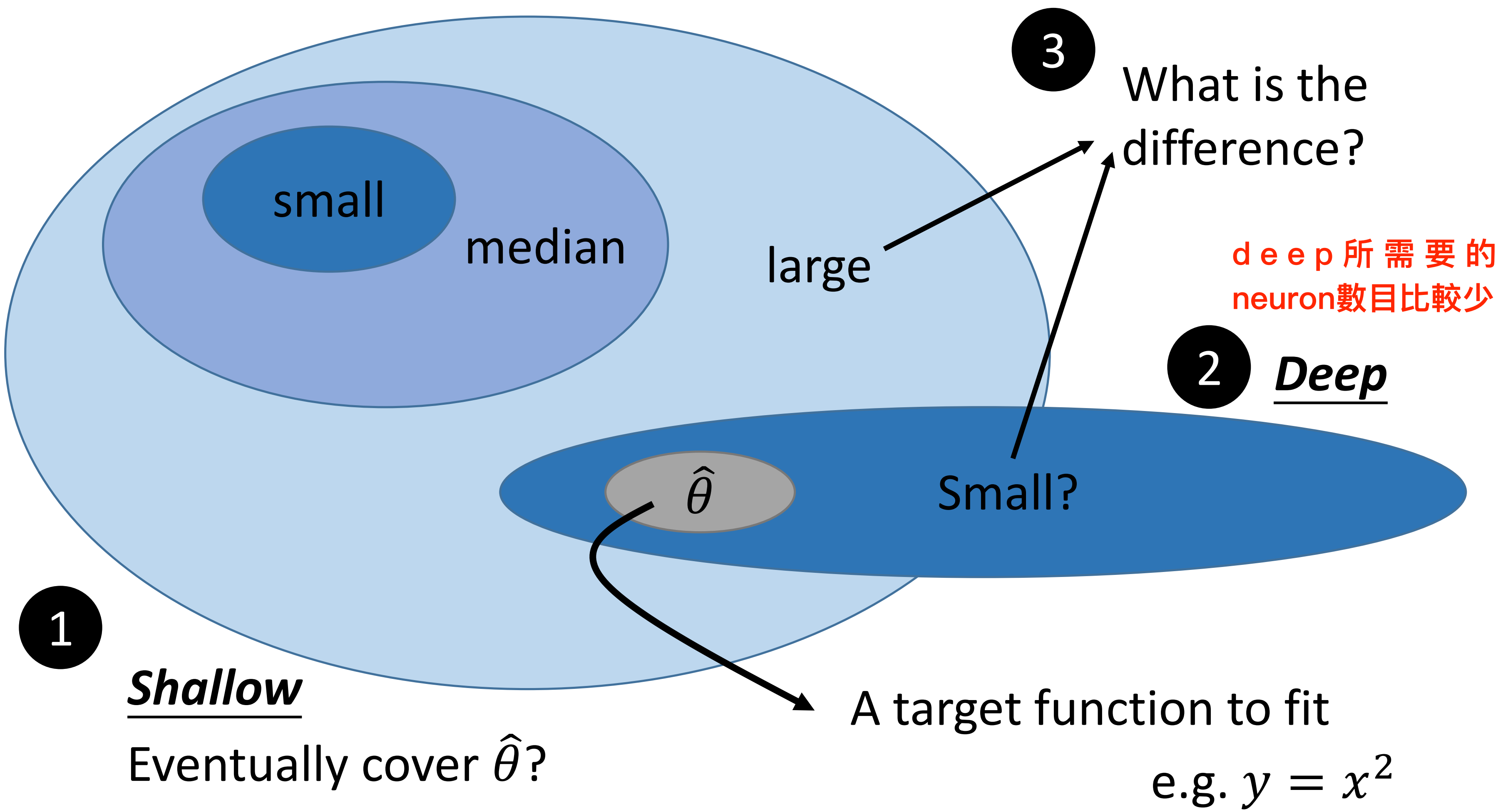
Outline

- Q1: Can shallow network fit any function? 可以
- Potential of deep
- Q2: How to use deep to fit functions?
- Q3: Is deep better than shallow?
- Review some related theories

ReLU as activation function

Outline

Notice: We do not discuss ***optimization*** and generalization today.



d e e p 所 需 要 的 neuron數目比較少

調整shallow的neuron樹木可以使得他fit函數

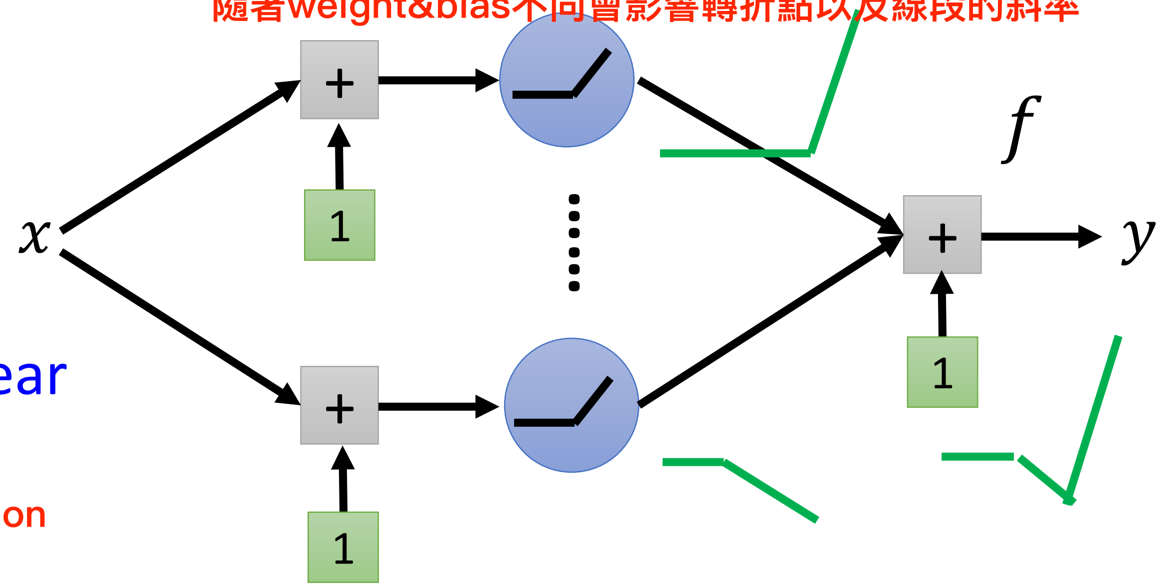
Can shallow network fit any function?

Universality

任何continuous function都可以用一層hidden layer來fit

- Given a ***shallow*** network structure with one hidden layer with ReLU activation and linear output

隨著weight&bias不同會影響轉折點以及線段的斜率



A piece-wise linear functions

由relu所組成的function

- Given a L-Lipschitz function f^*
    - How many neurons are needed to approximate f^*?

Universality

- Given a L-Lipschitz function f^*
  - How many neurons are needed to approximate f^*?

變化比較快的線比較不像1–Lipschitz function，因為她應該表現平滑

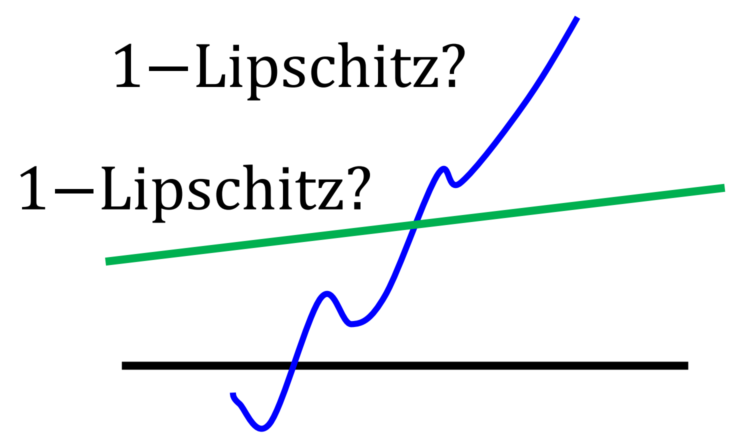
<u>L-Lipschitz Function</u> (smooth)

output變化會被input變化所bounded

$$\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$$

Output change   Input change

L=1 for "$1 - Lipschitz$"



Universality

$$\max_{0\le x\le 1}|f(x)-f^*(x)|\le\varepsilon$$

$$\sqrt{\int_0^1|f(x)-f^*(x)|^2\,dx}\le\varepsilon$$

max的條件滿足的話積分 給也就自動被滿足了，
注意範圍是0–1

- Given a L-Lipschitz function f^*
    - How many neurons are needed to approximate f^*?

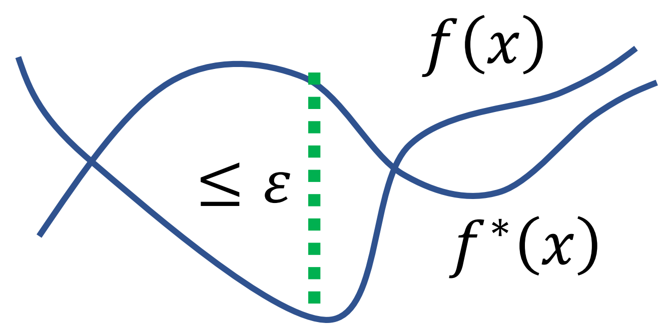
$f\in N(K)$ → The function space defined by the network with K neurons.

Given a small number $\varepsilon>0$

What is the number of K such that

Exist $f\in N(K)$, $\max\limits_{0\le x\le 1}|f(x)-f^*(x)|\le\varepsilon$

The difference between $f(x)$ and $f^*(x)$ is smaller than ε.



Universality

- L-Lipschitz function f^*

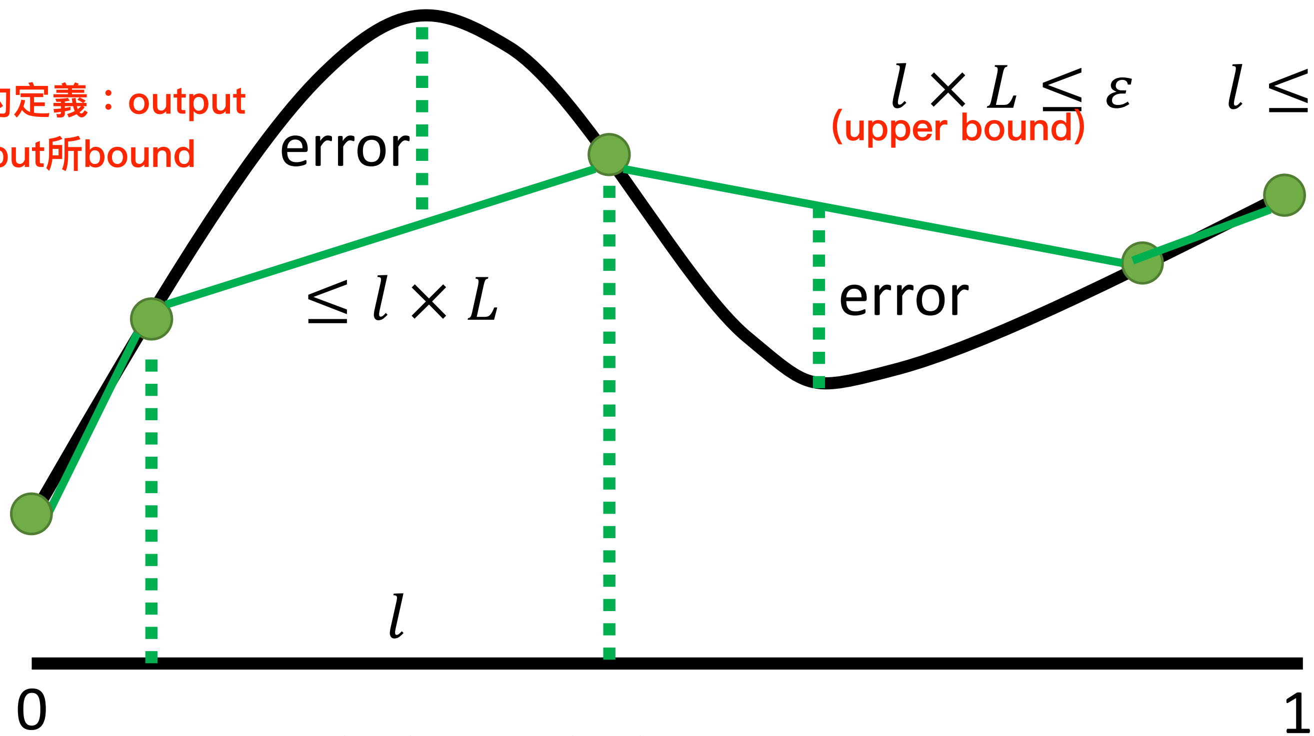
All the functions in $N(K)$ are piecewise linear.

Approximate f^* by a piecewise linear function f

How to make the errors $\leq \varepsilon$

$l \times L \leq \varepsilon$ (upper bound) $\quad l \leq \varepsilon / L$

L–Lipschitz的定義：output 的變化會被input所bound



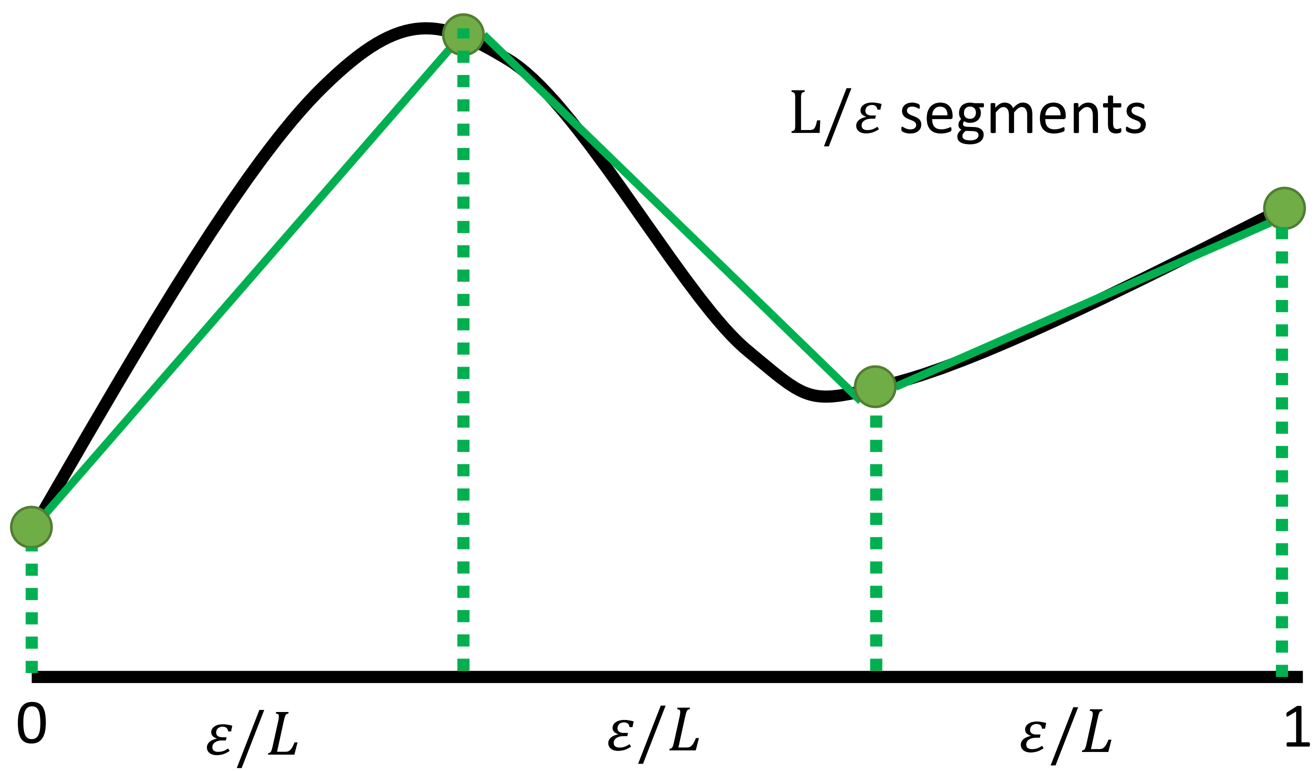
$$\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$$

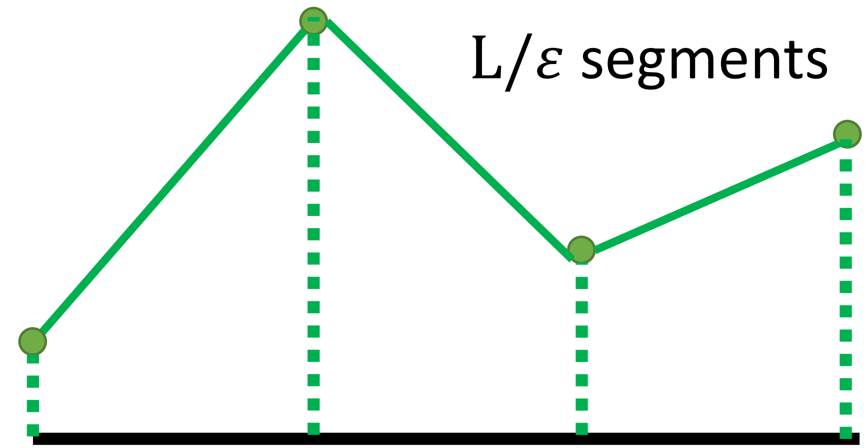
給訂一個區間長度為l，找出其最高點跟最低點，則距離差距一定<l，又因為L–Lipschitz，因此最大誤差<lxL

Universality

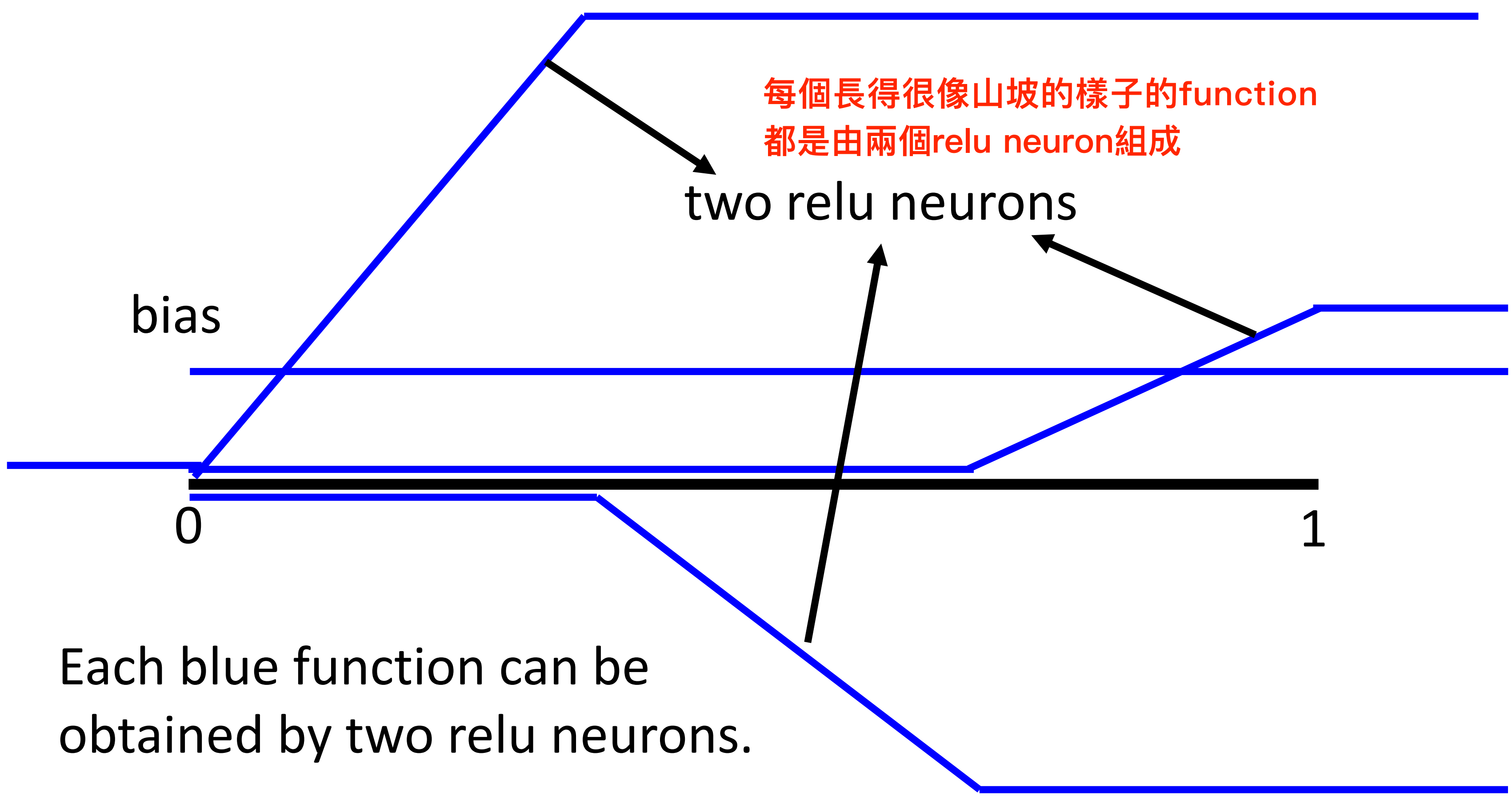
- L-Lipschitz function f^*

How to make a 1 hidden layer relu network have the output like green curve?

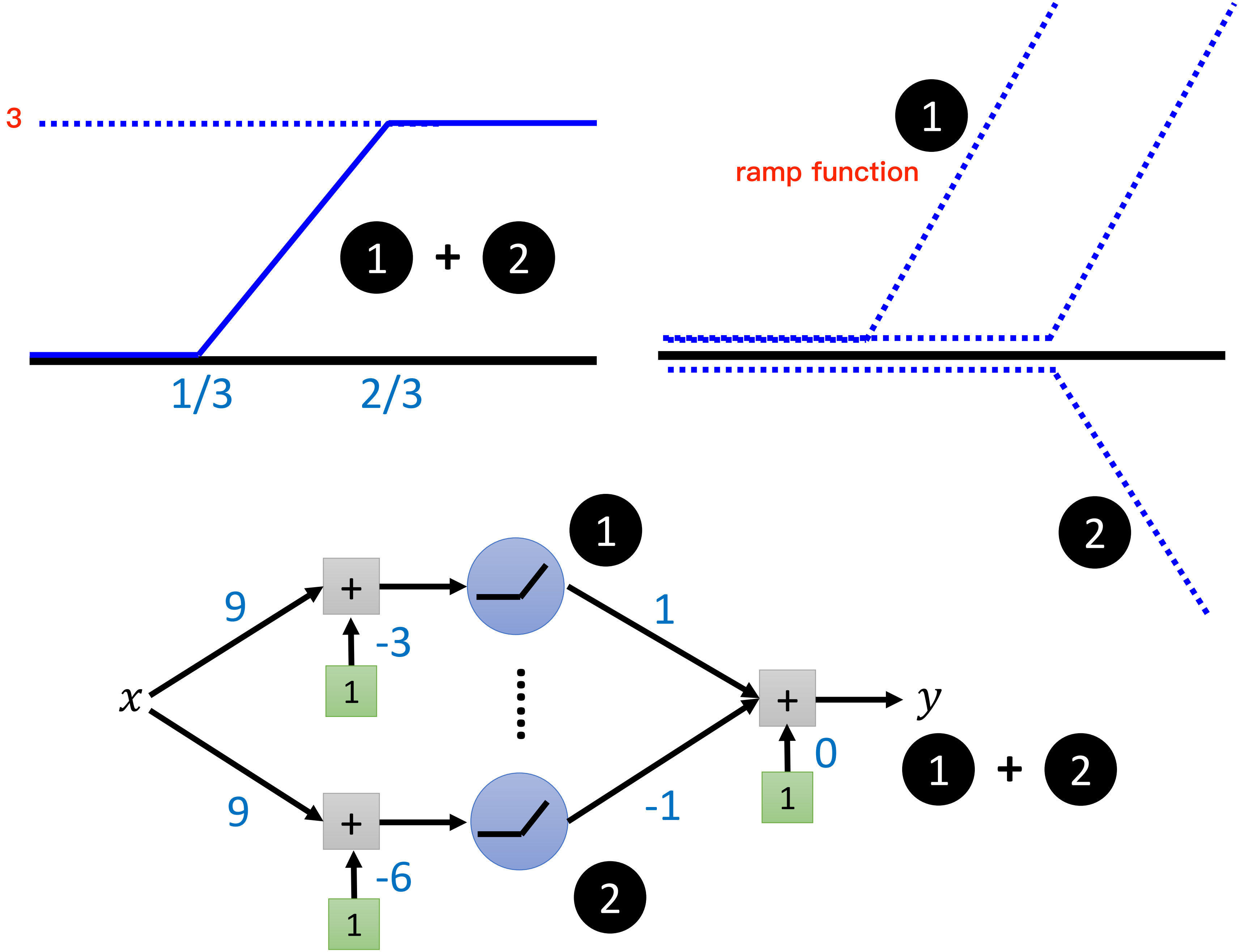


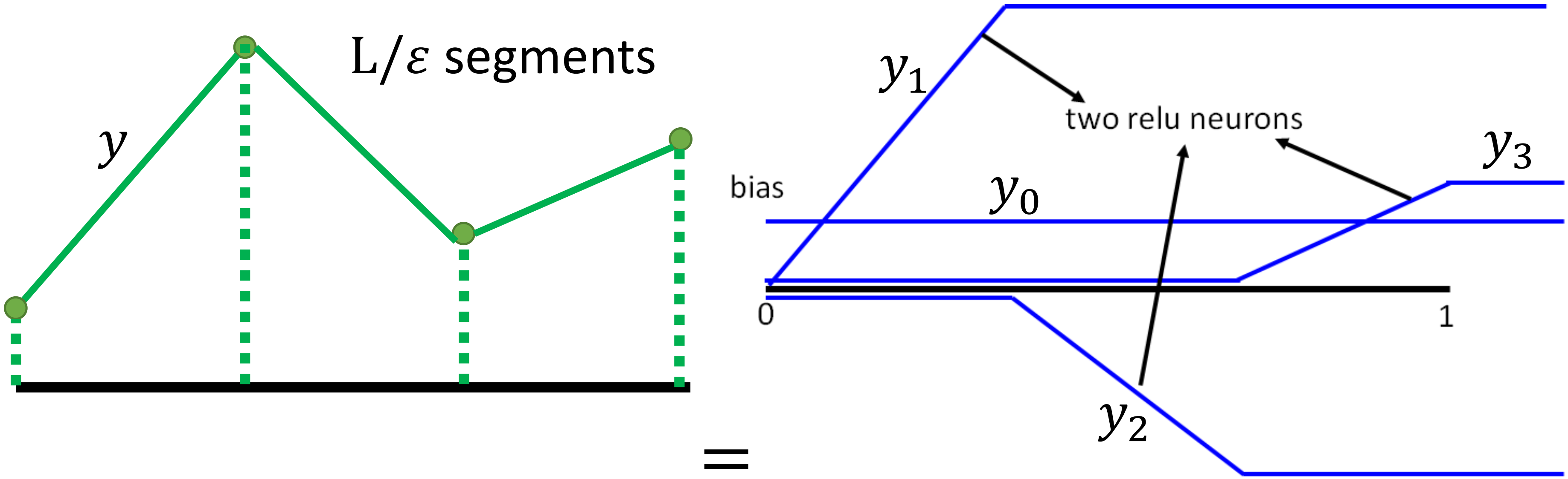


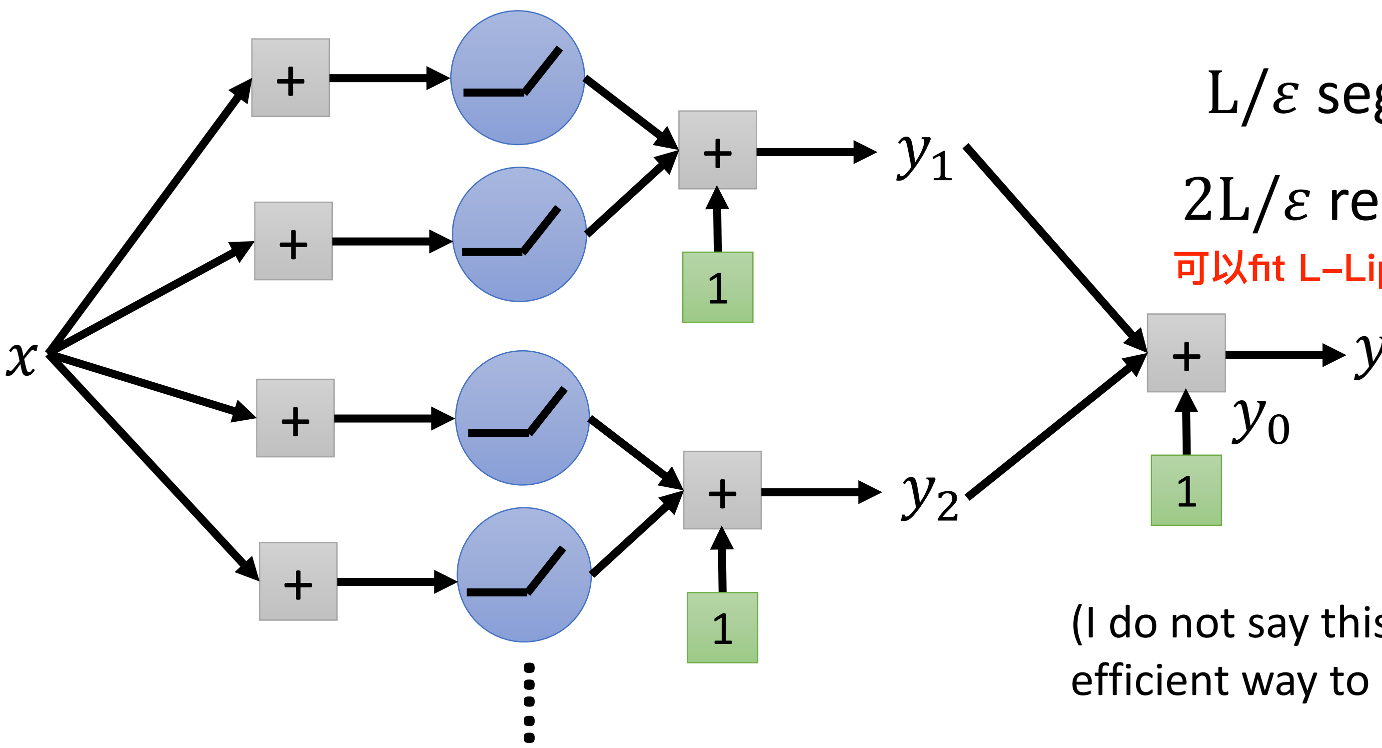
The summation of the blue functions is the green one.



Each blue function can be obtained by two relu neurons.


3
1/3
2/3
1 + 2
1
ramp function
2
9
-3
9
-6
x
1
-1
0
y
1 + 2





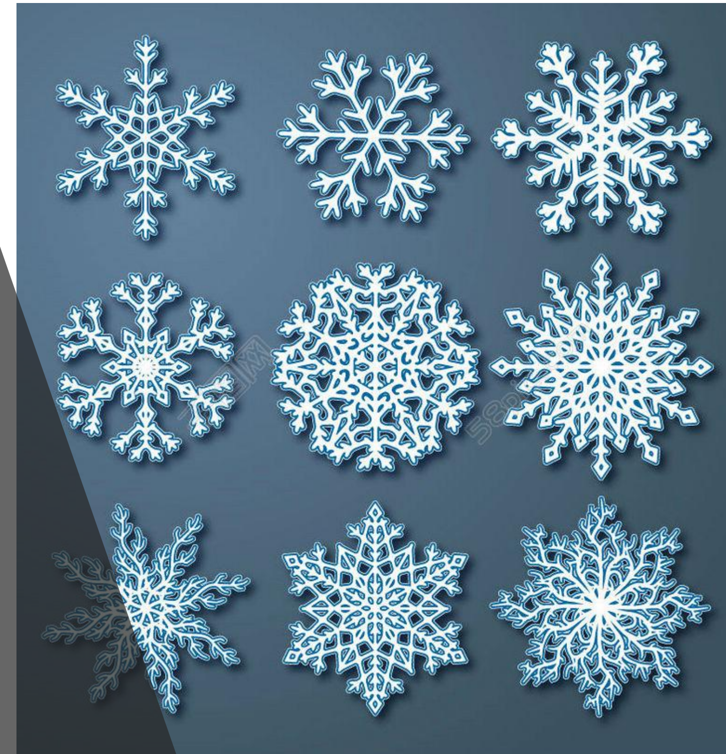
L/ε segments

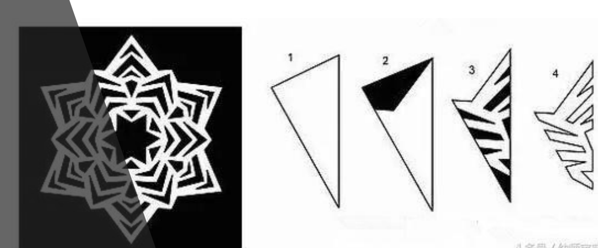
2L/ε relu neurons

可以fit L–Lipsitchz function

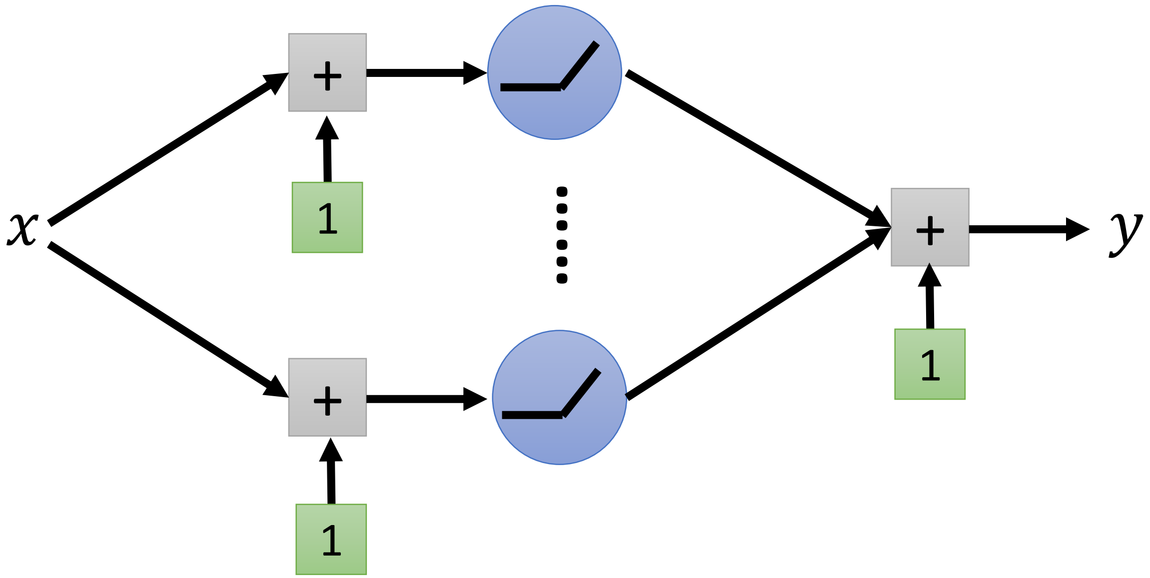
(I do not say this is the most efficient way to use the neurons.)

Potential of deep





Why we need deep?



neuron number不同

Yes, shallow network can represent any function.

However, using deep structure is more effective.

Analogy – Programming

- Solve any problem by two lines (shallow)
  - Input = K
  - Line 1: row no. = MATCH_KEY(K)
  - Line 2: Output the value at row no.

Input (key)	Output (value)
A	A’
B	B’
C	C’
D	D’
……	……

ＳＶＭ作法與上面有點相似

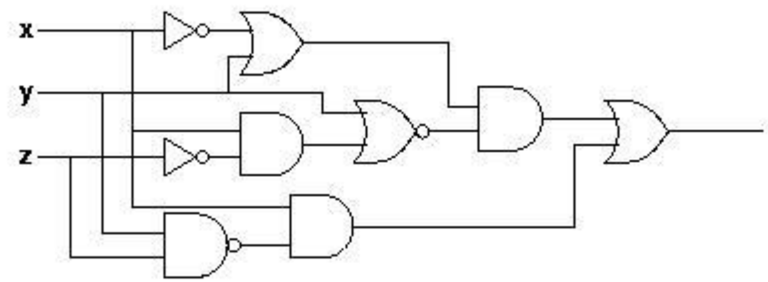
- Considering SVM with kernel

$$y = \sum_{n} \alpha_n K(x^n, x)$$

把每筆data跟input x計算相似度後乘上常數後得到結果

- Using multiple steps to solve problems is more efficient (deep)

Analogy



Logic circuits

- Logic circuits consists of **gates**
- **A two layers of logic gates** can represent **any Boolean function.**
- Using multiple layers of logic gates to build some functions are much simpler

→ less gates needed

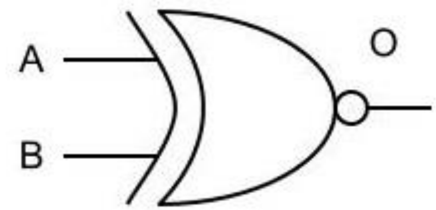
Neural network

- Neural network consists of **neurons**
- **A hidden layer network** can represent **any continuous function.**
- Using multiple layers of neurons to represent some functions are much simpler

→ less neurons

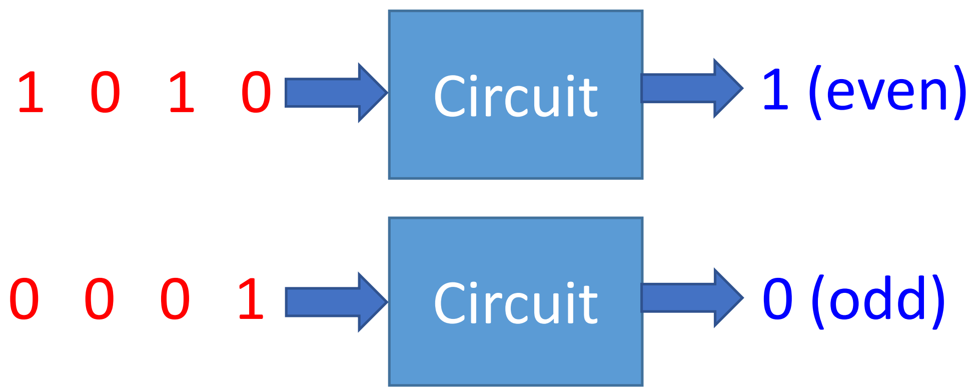
This page is for EE background.

Analogy



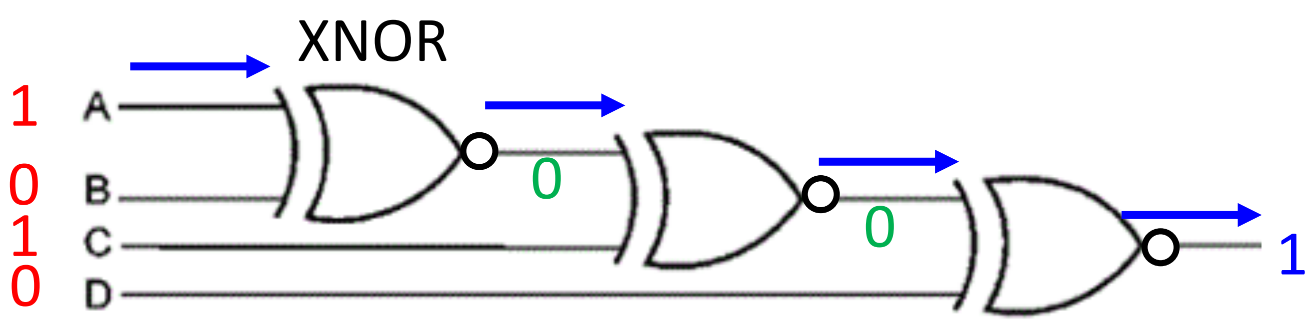
A	B	O
0	0	1
0	1	0
1	0	0
1	1	1

- E.g. ***parity check***



For input sequence with d bits,

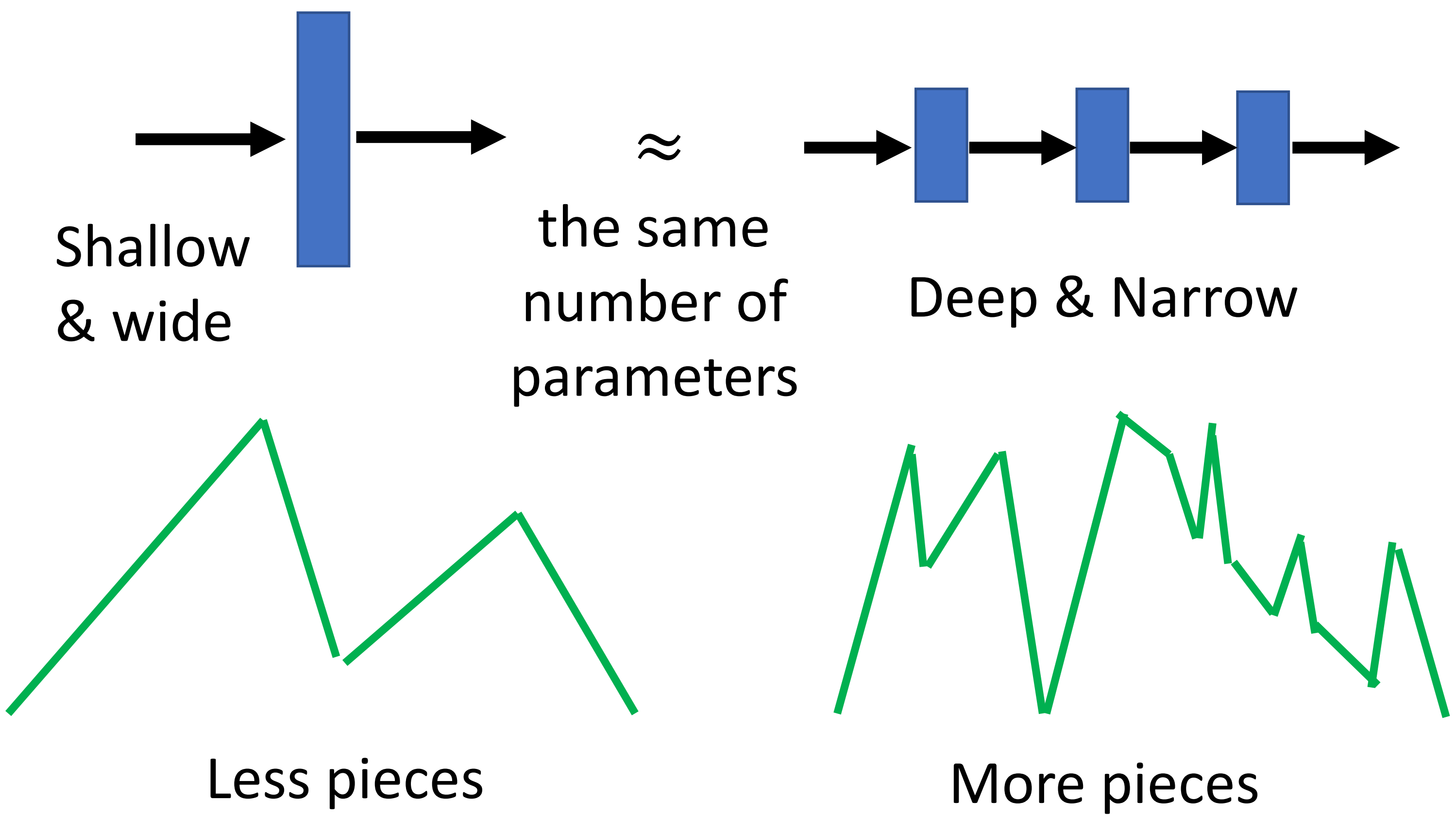
Two-layer circuit need $O(2^d)$ gates.



With multiple layers, we need only O(d) gates.

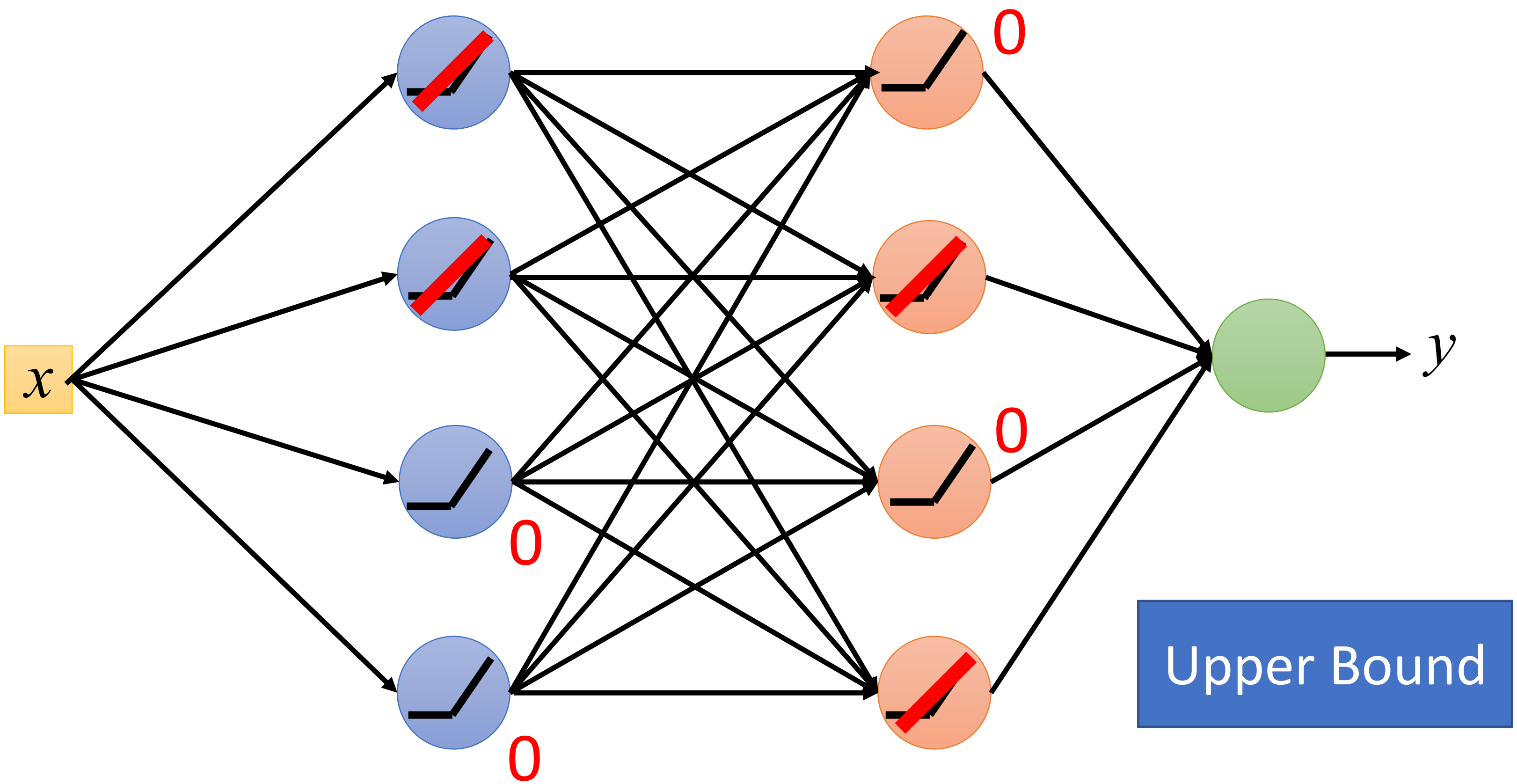
Why we need deep?

- ReLU networks can represent piecewise linear functions



最多可以組成多少piece wise的 linear function

Upper Bound of Linear Pieces



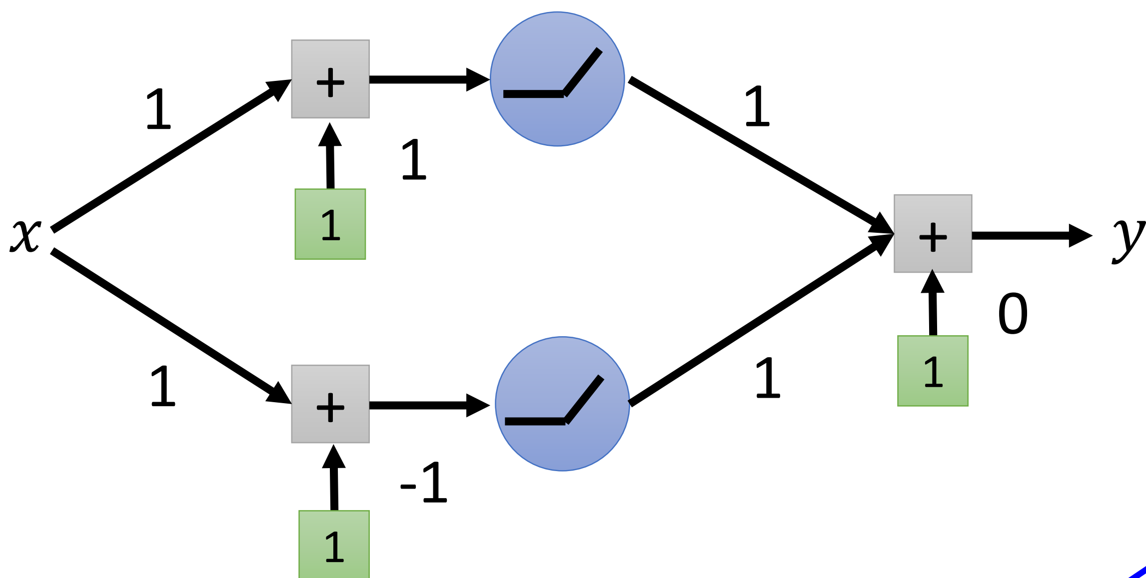
Each “activation pattern” defines a linear function.

N neurons → 2^N “activation patterns” → 2^N “linear pieces”

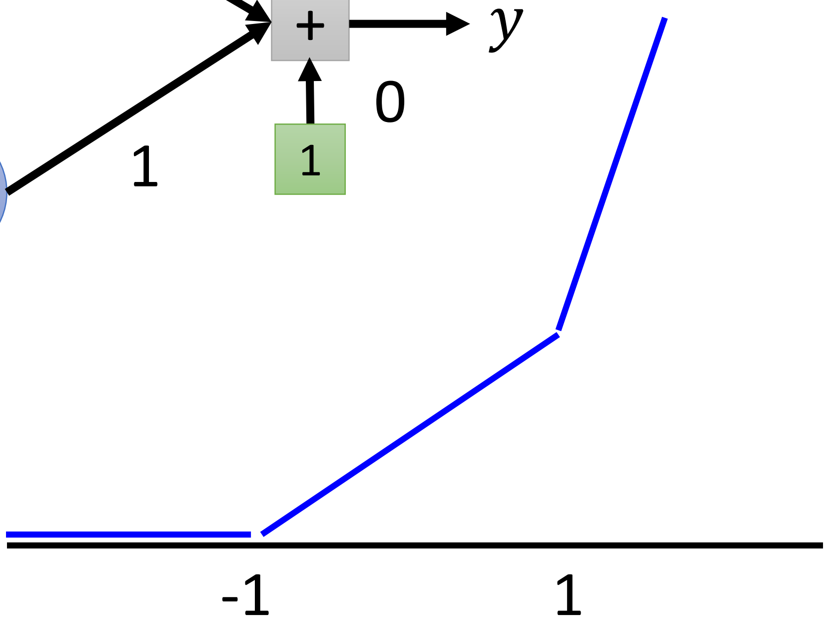
Upper bound: 每個relu可以有兩個model：0或是linear，因此總共有2的n次方個linear pieces

Upper Bound of Linear Pieces

- Not all the “activation patterns” available

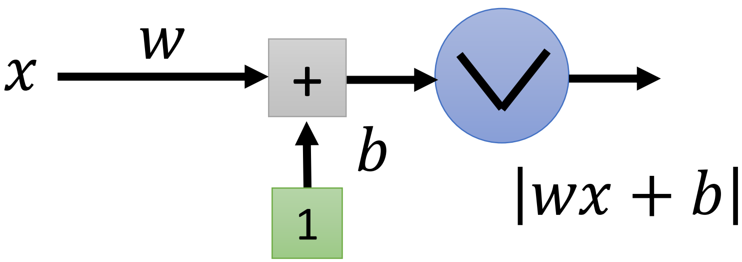


In shallow network, each neuron only provides one linear piece.

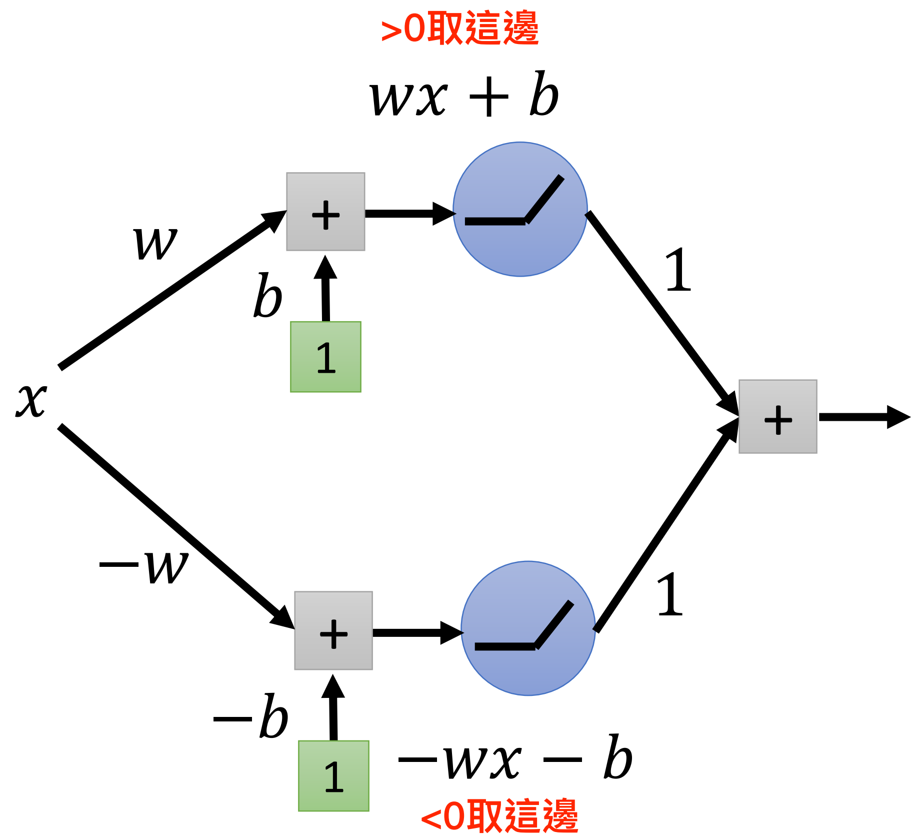


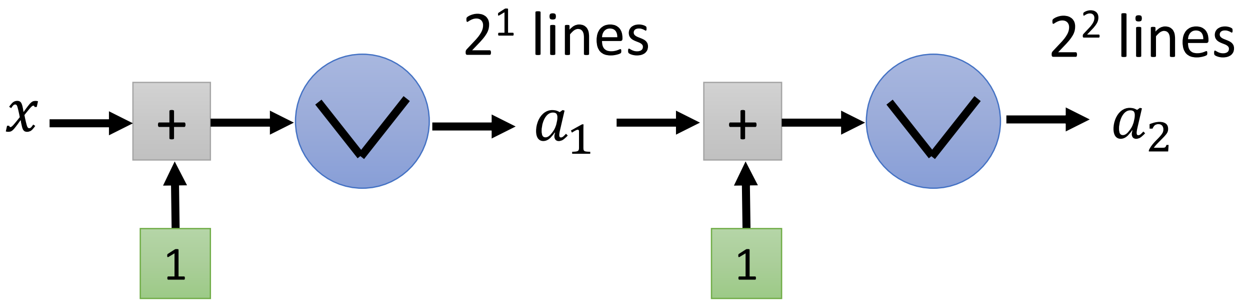
以這個例子來看最多只能組成三種pattern

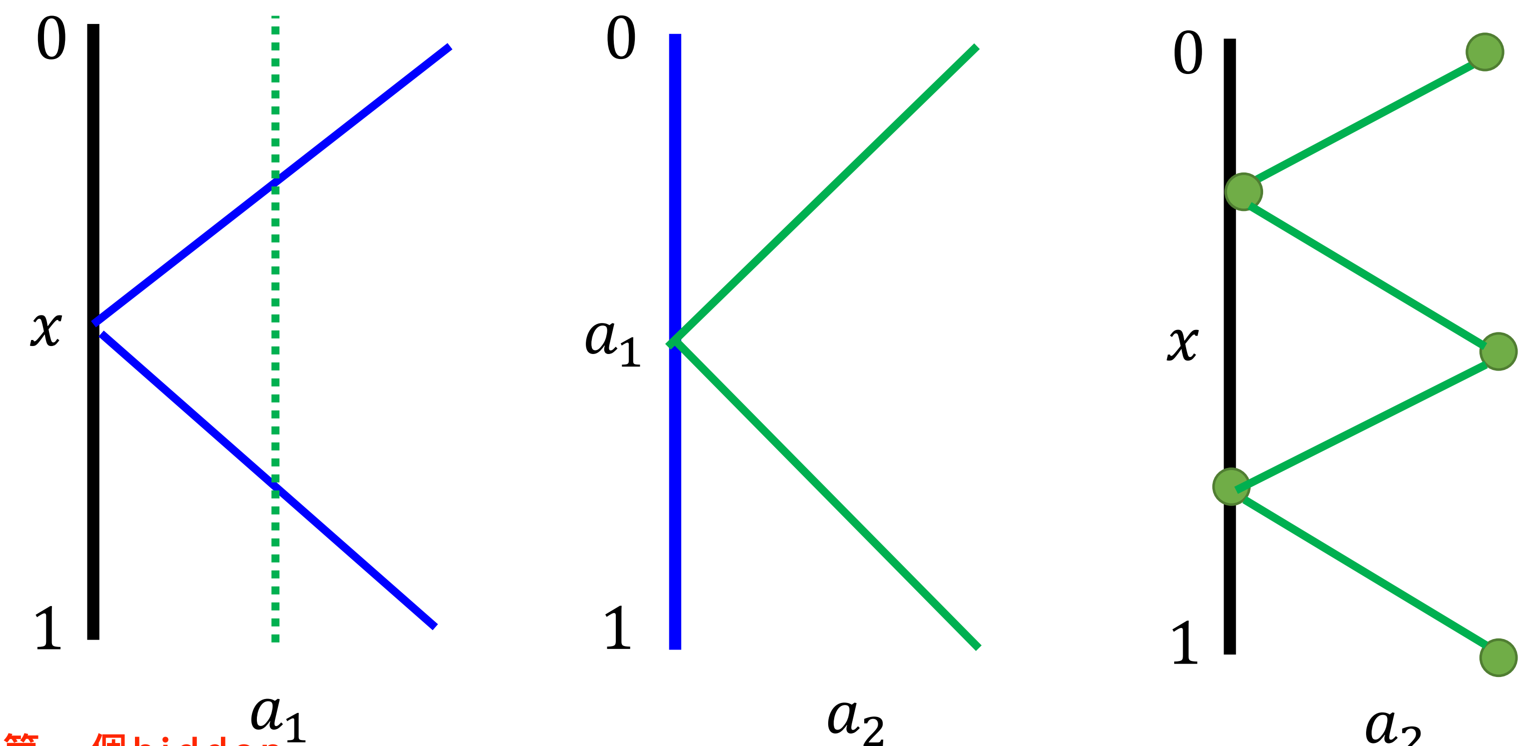
Abs Activation Function



Use two relu to implement
an abs activation function

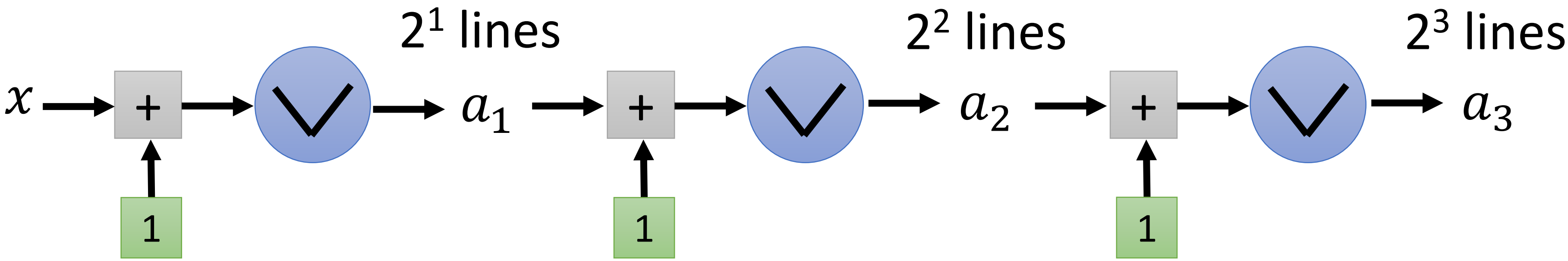


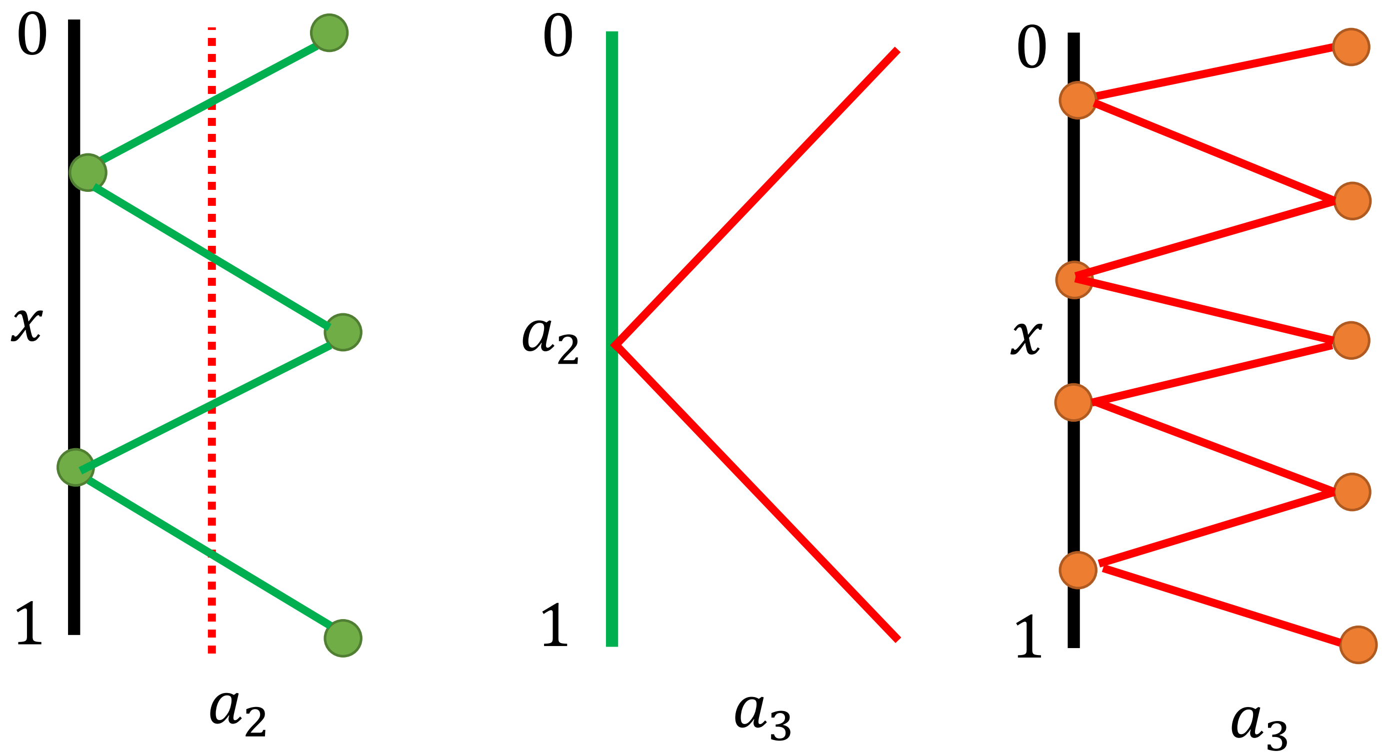




假設第一個hidden layer做上圖的事情

Each node added ➡ The regions are twice.

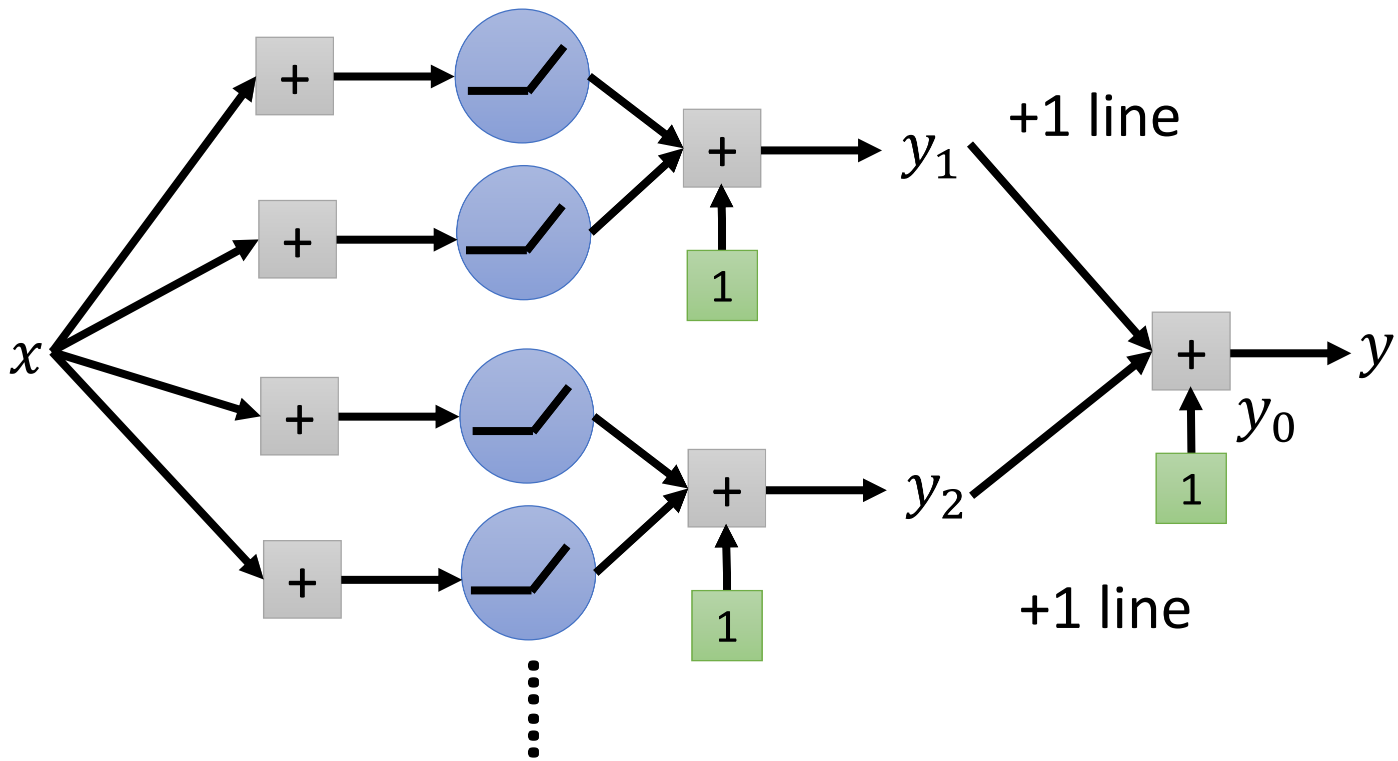




假設要產生8個pieces的function

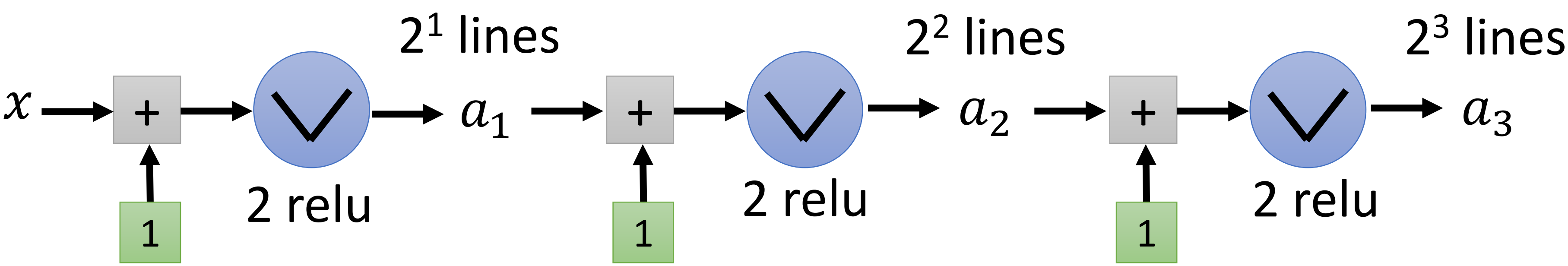
Shallow

需要16個neuron



Deep

需要3個neuron



Lower Bound of Linear Pieces

If K is width, H is depth

We can have at least K^H pieces

Depth has much larger influence than depth.

Razvan Pascanu, Guido Montufar, Yoshua Bengio, "On the number of response regions of deep feed forward networks with piece-wise linear activations", ICLR, 2014

Guido F. Montufar, Razvan Pascanu, Kyunghyun Cho, Yoshua Bengio, "On the Number of Linear Regions of Deep Neural Networks", NIPS, 2014

Raman Arora, Amitabh Basu, Poorya Mianjy, Anirbit Mukherjee, "Understanding Deep Neural Networks with Rectified Linear Units", ICLR 2018

Thiago Serra, Christian Tjandraatmadja, Srikumar Ramalingam, "Bounding and Counting Linear Regions of Deep Neural Networks", arXiv, 2017

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, Jascha Sohl-Dickstein, On the Expressive Power of Deep Neural Networks, ICML, 2017

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, Jascha Sohl-Dickstein, On the Expressive Power of Deep Neural Networks, ICML, 2017

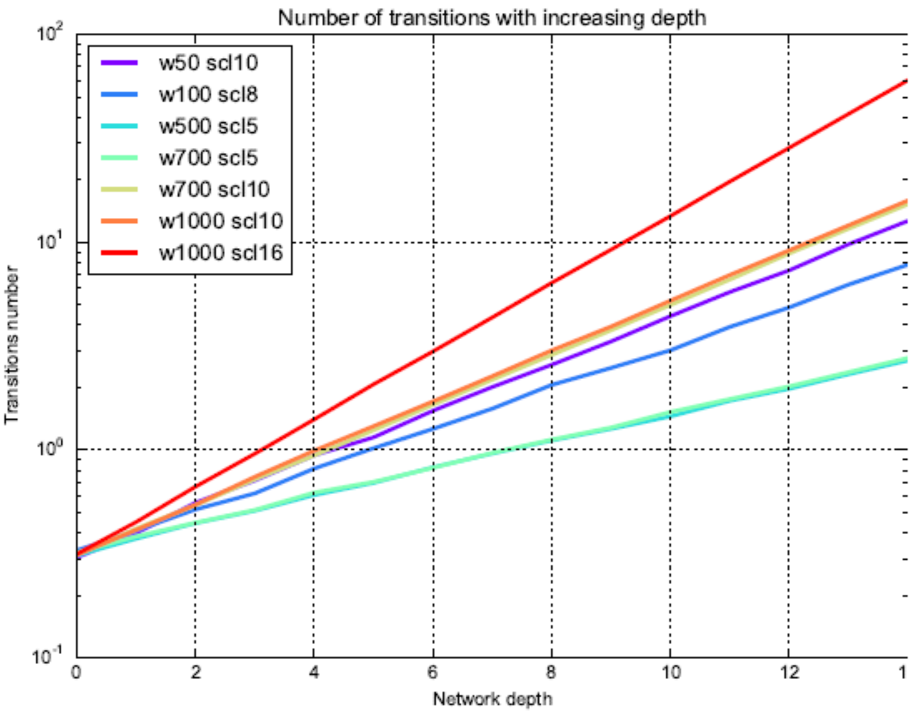
Experimental Results

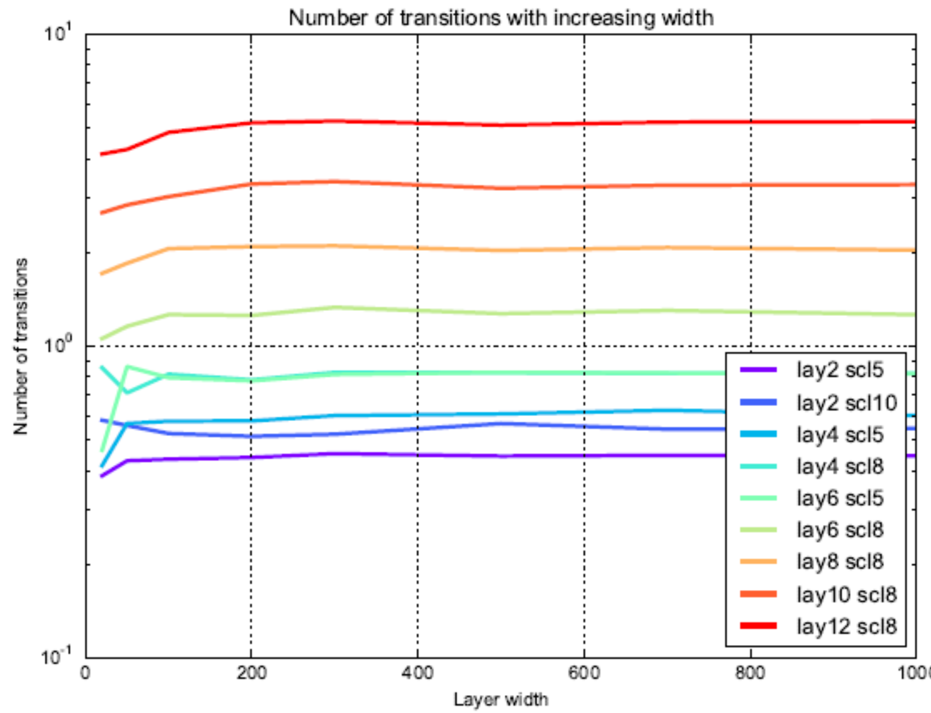
可以看得出是對稱的，有固定的pattern的，呼應剛剛所說的activation pattern

(MNIST)

縱軸代表所經過的piece數目，注意單位是exponential

固定depth調整width



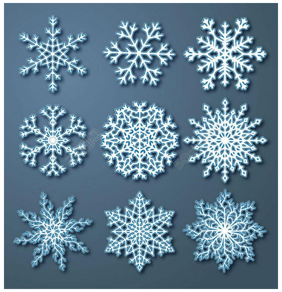


input

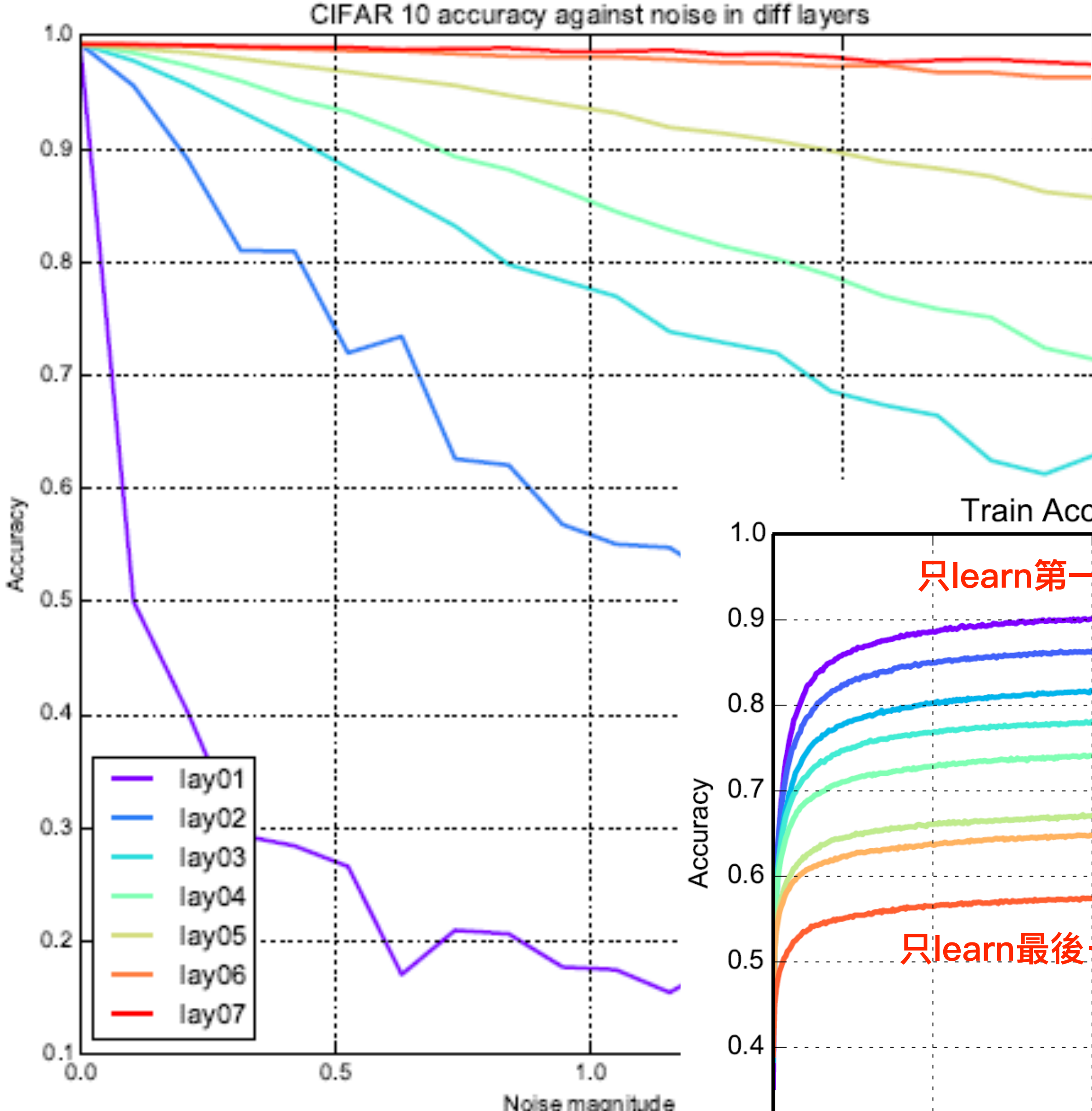
第二層project到2維

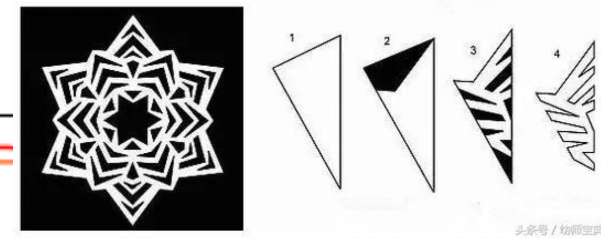
第三層project到2維

第層四project到2維

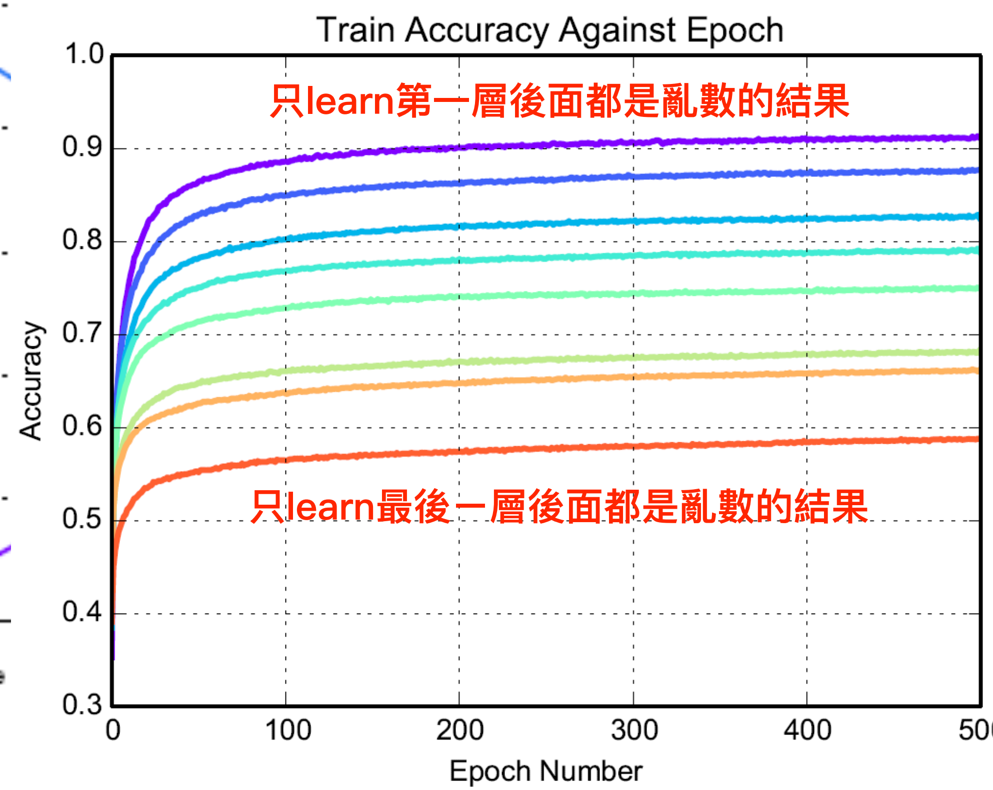


針對各層加入noise後對其正確率的影響



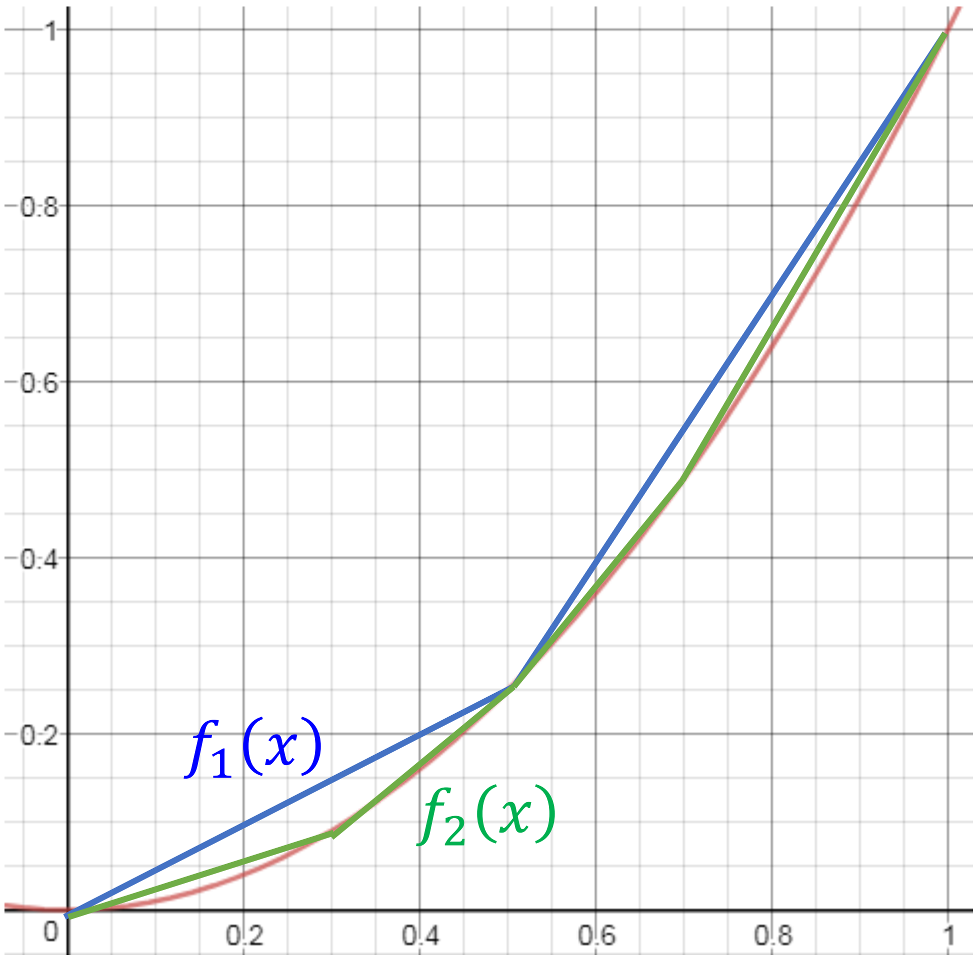


low layer的參數是比較重要的



How much is deep better than shallow?

$f(x) = x^2$



Fit the function by equally spaced linear pieces

$f_m(x)$: a function with 2^m pieces

$$\max_{0 \le x \le 1} |f(x) - f_m(x)| \le \varepsilon$$

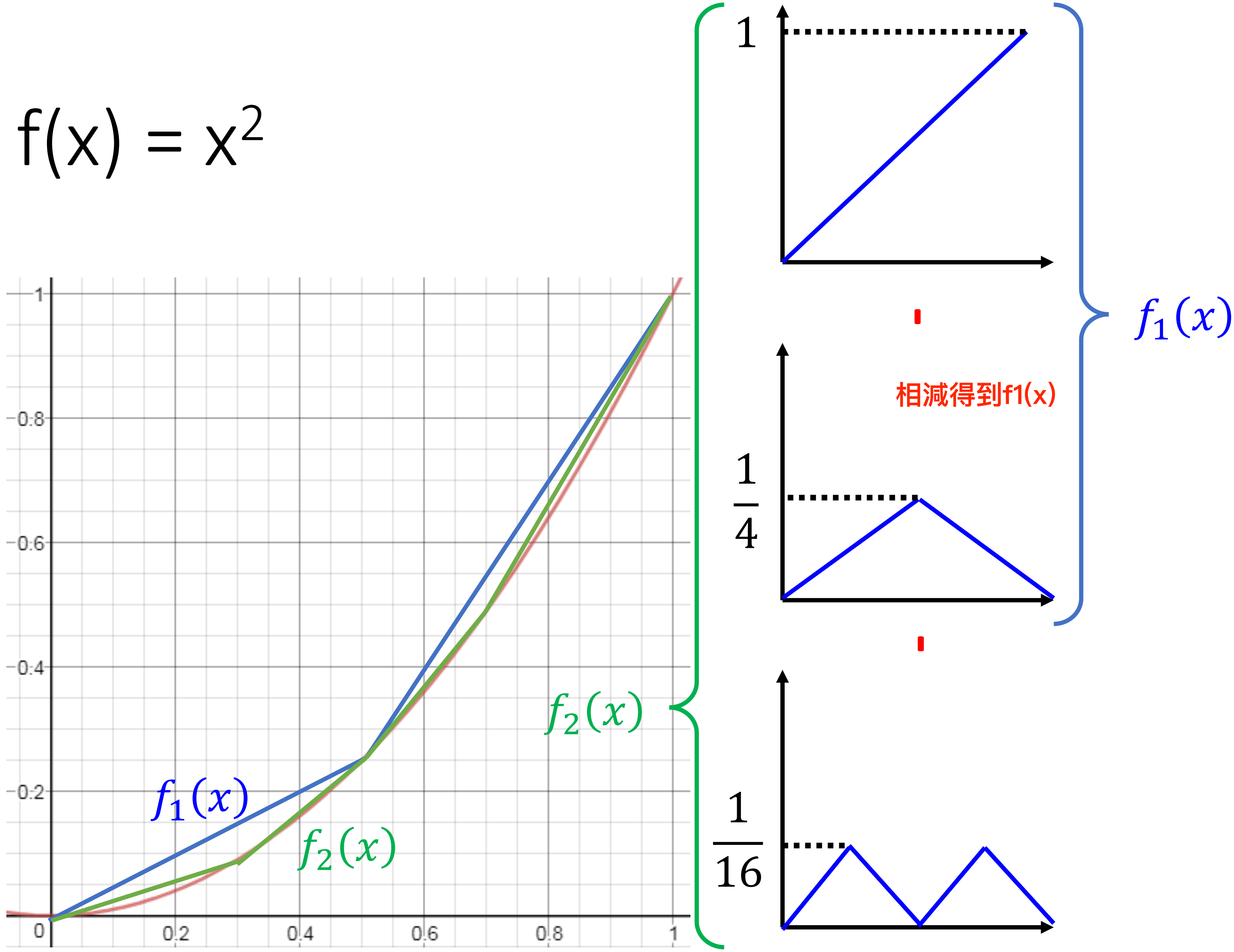
What is the minimum m?

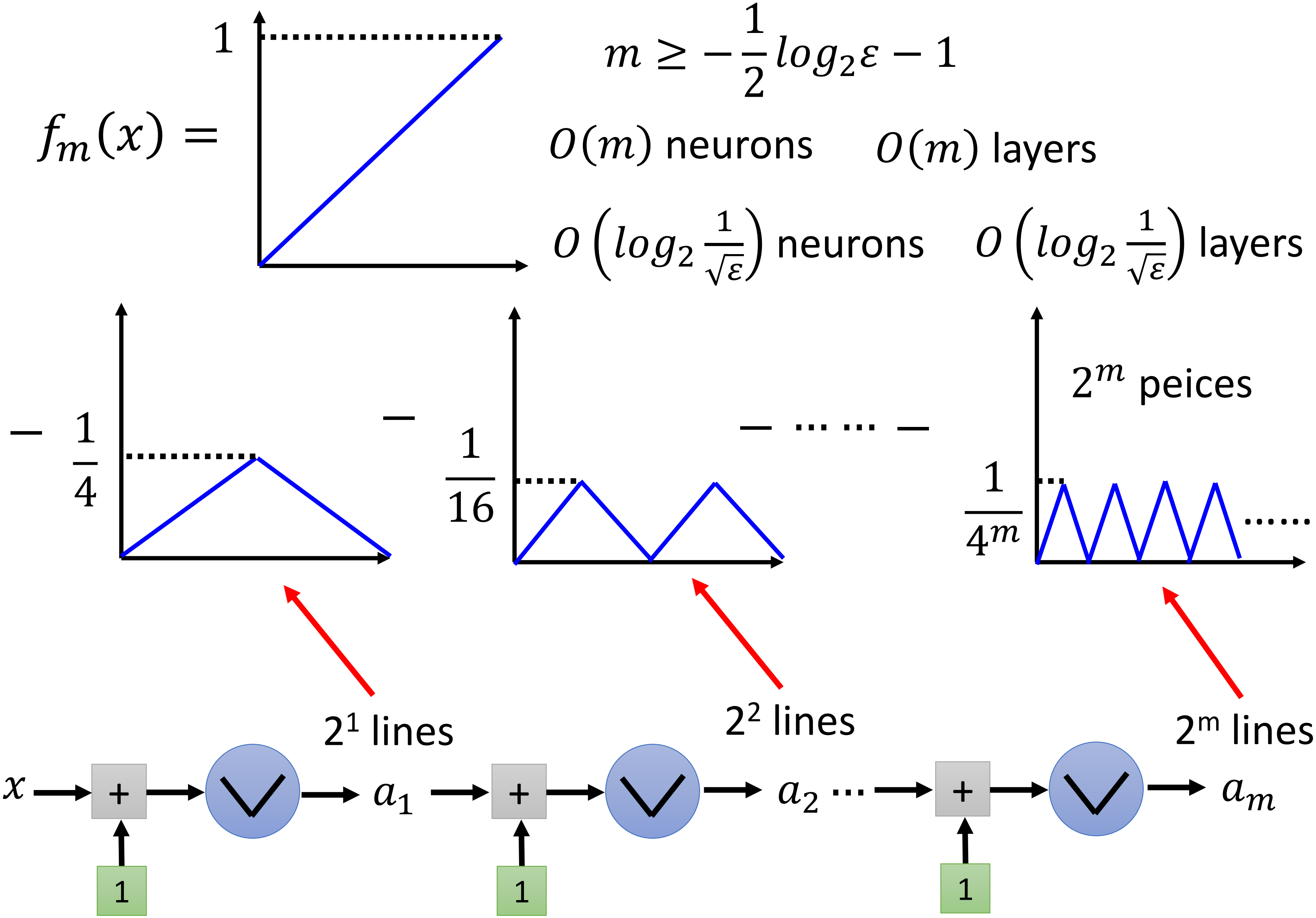
$$m \geq -\frac{1}{2} log_2 \varepsilon - 1$$

$$2^m \geq \frac{1}{2} \frac{1}{\sqrt{\varepsilon}} \text{ pieces}$$

Shallow: $O\left(\frac{1}{\sqrt{\varepsilon}}\right)$ neurons

$f(x) = x^2$

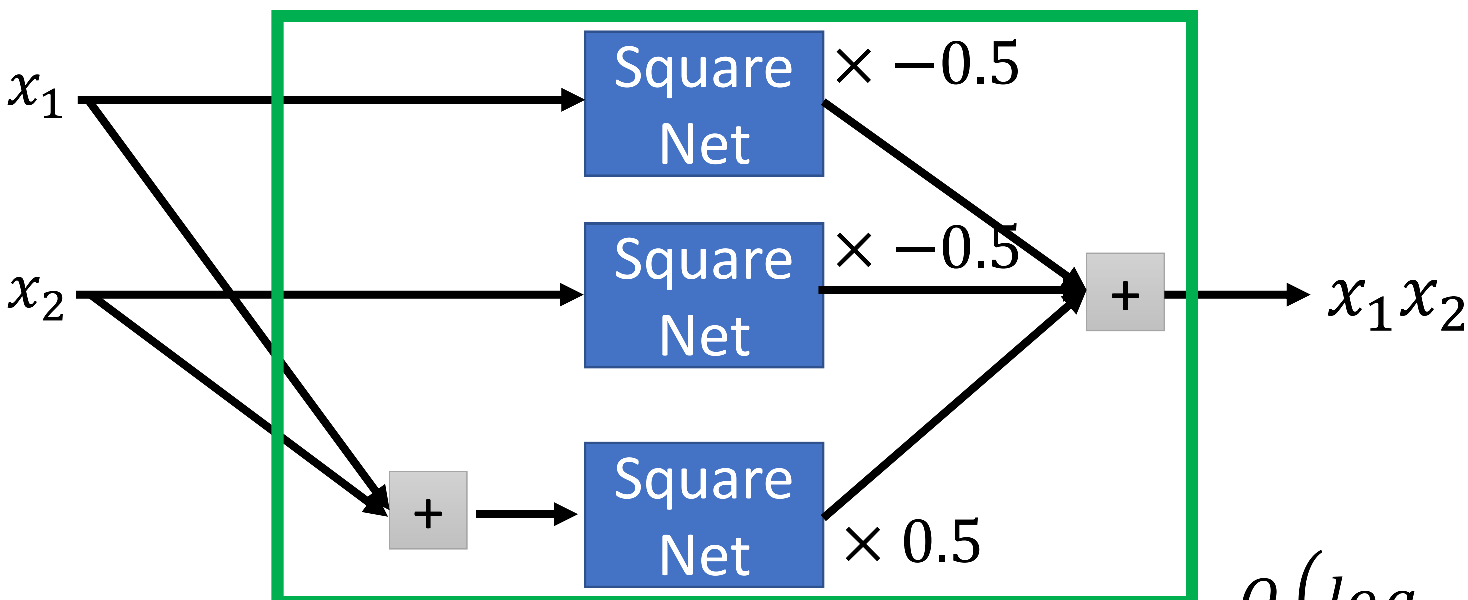



$f_m(x) =$
1
$m \geq -\frac{1}{2} log_2 \varepsilon - 1$
$O(m)$ neurons
$O(m)$ layers
$O\left(log_2 \frac{1}{\sqrt{\varepsilon}}\right)$ neurons
$O\left(log_2 \frac{1}{\sqrt{\varepsilon}}\right)$ layers
$- \frac{1}{4}$
$- \frac{1}{16}$
$- \cdots \cdots -$
$\frac{1}{4^m}$
2^m peices
2^1 lines
2^2 lines
2^m lines
x
a_1
$a_2 \cdots$
a_m
+
1

Why care about $y = x^2$?

$O\left(log_2 \frac{1}{\sqrt{\varepsilon}}\right)$ neurons

$$y = x_1 x_2$$
$$= \frac{1}{2}\left((x_1 + x_2)^2 - x_1^2 - x_2^2\right)$$



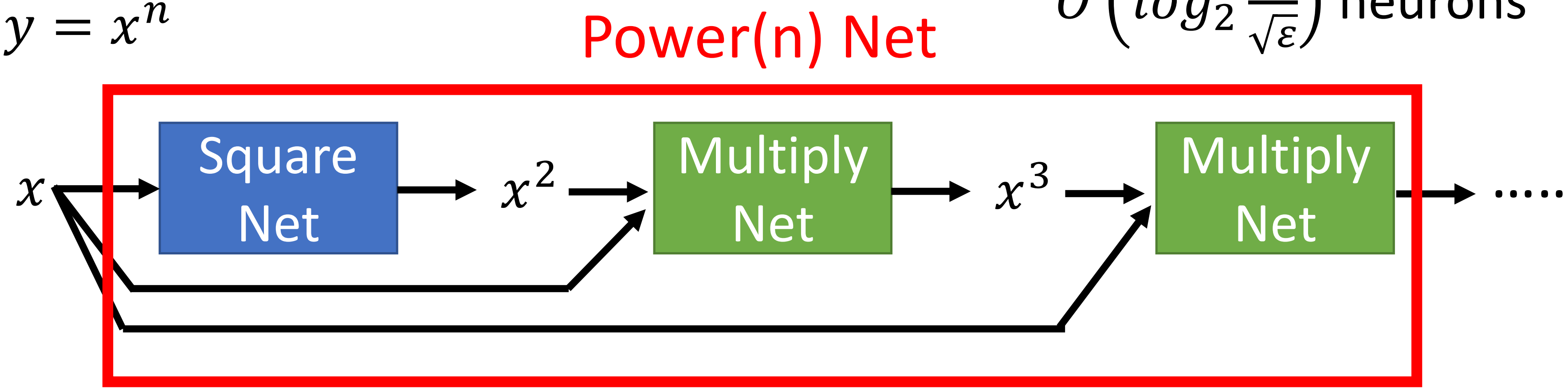
Multiply Net

$O\left(log_2 \frac{1}{\sqrt{\varepsilon}}\right)$ neurons

Polynomial

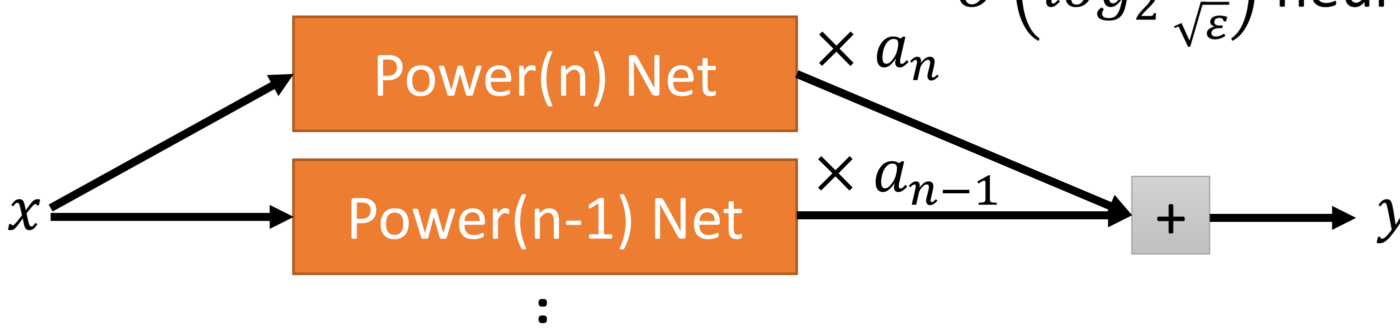
$y = x^n$

$O\left(log_2 \frac{1}{\sqrt{\varepsilon}}\right)$ neurons



$y = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$

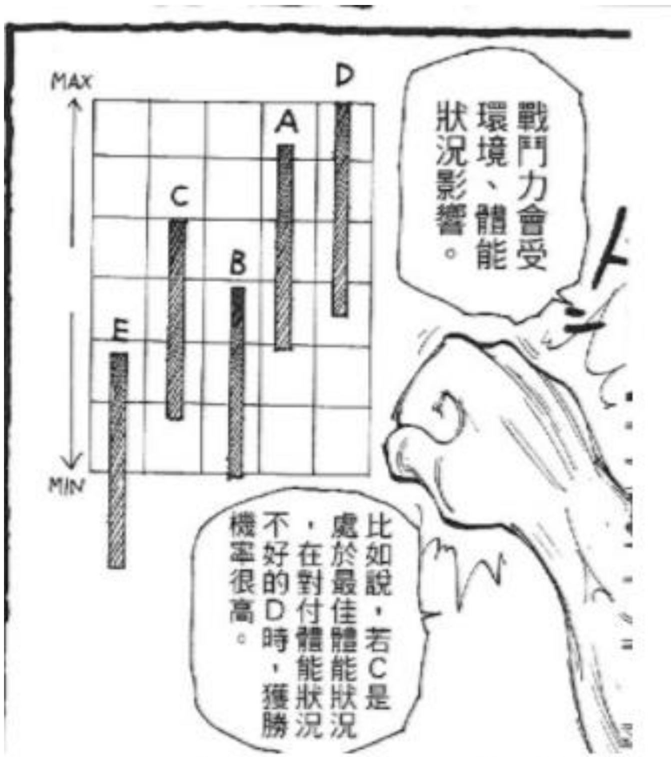
$O\left(log_2 \frac{1}{\sqrt{\varepsilon}}\right)$ neurons



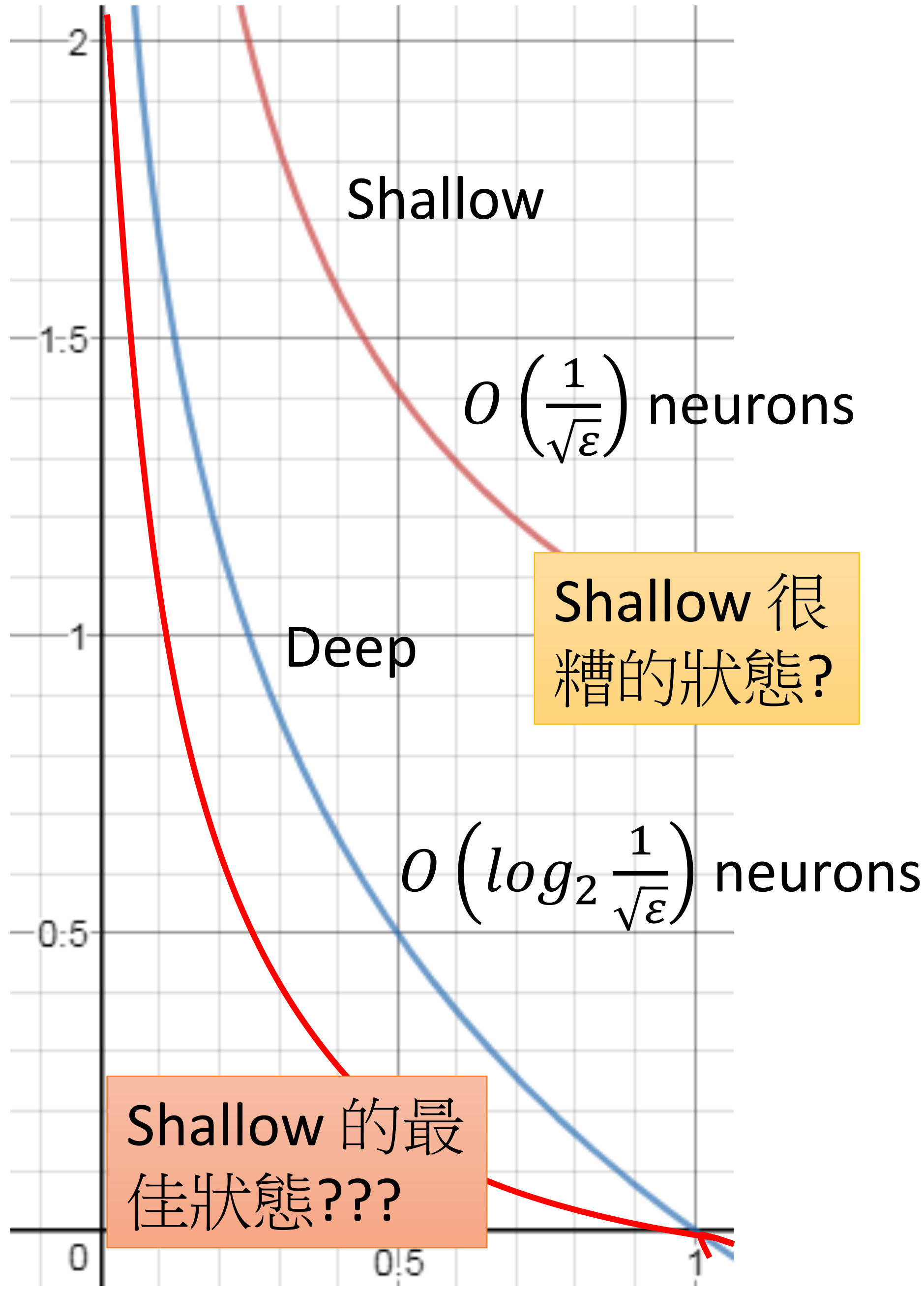
Use polynomial function to fit other functions.

Deep v.s. Shallow

This is not sufficient to show the power of deep.



(獵人第二十卷)



Is Deep better than Shallow?

Best of Shallow

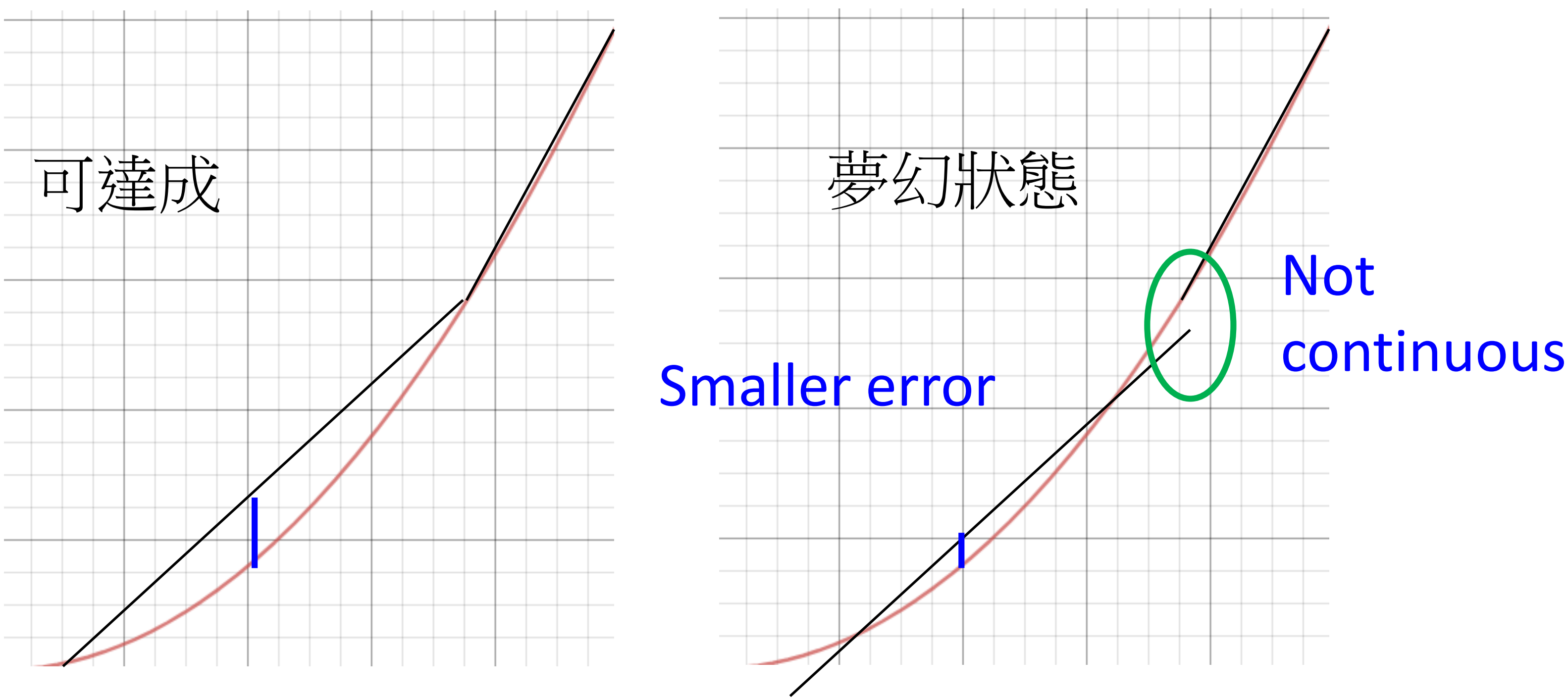
$$\max_{0 \le x \le 1} |f(x) - f^*(x)| \le \varepsilon$$

$$\Downarrow$$

$$\sqrt{\int_0^1 |f(x) - f^*(x)|^2 \, dx} \le \varepsilon$$

Use Euclidean

- A relu network is a piecewise linear function.
- Using the least pieces to fit the target function.



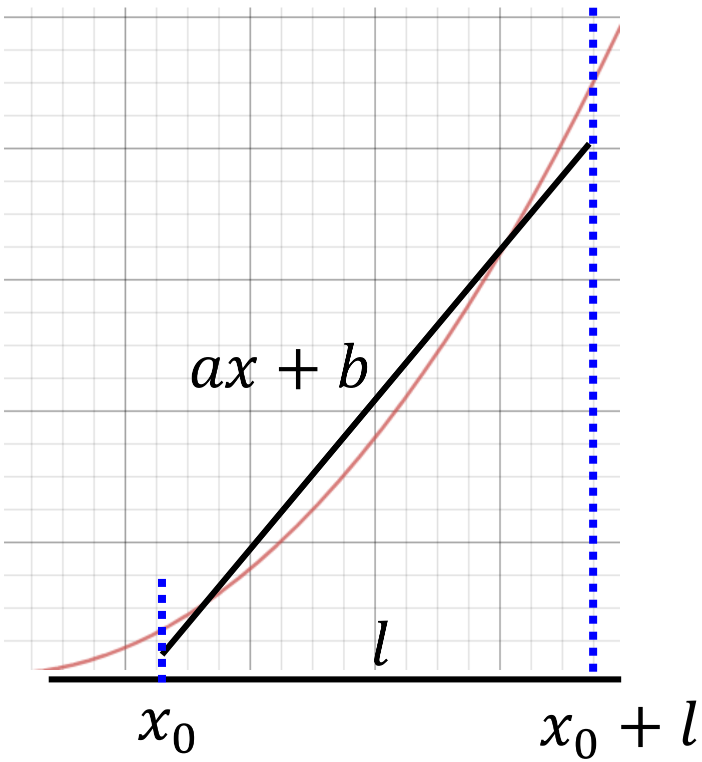
The lines do not have to connect the end points.

Best of Shallow

$$\sqrt{\int_0^1 |f(x) - f^*(x)|^2\, dx} \leq \varepsilon$$

Use Euclidean

- Given a piece, what is the smallest error



$$e^2 = \int_{x_0}^{x_0+l} \left(x^2 - (ax + b)\right)^2 dx$$

Find a and b to minimize e^2

The minimum value of e^2 is $\frac{l^5}{180}$

Warning of Math

Intuition

$$e^2 = \int_{x_0}^{x_0+l} \left(x^2 - (ax + b\ )\right)^2 dx$$

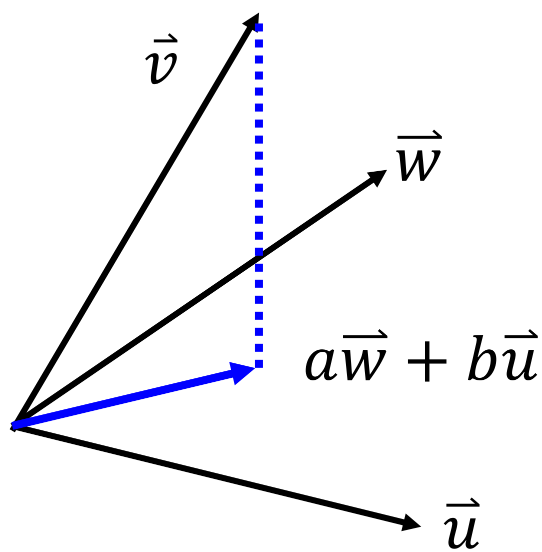
$$f_v = x^2 \qquad f_w = x \qquad f_u = 1$$

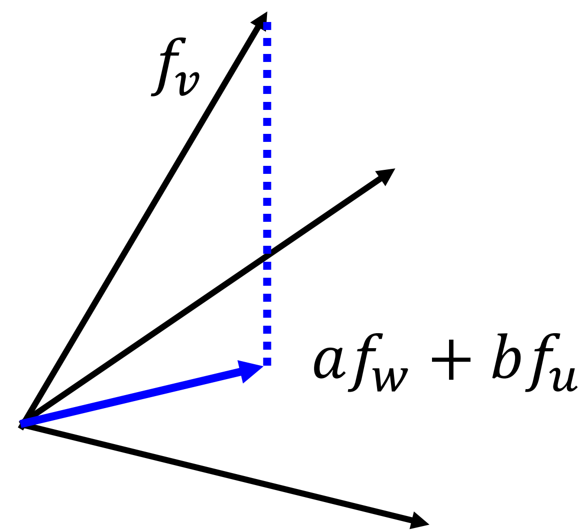
Minimize

$$\|\vec{v} - (a\vec{w} + b\vec{u})\|^2$$

Minimize

$$\|f_v - (af_w + bf_u)\|^2$$





End of Warning

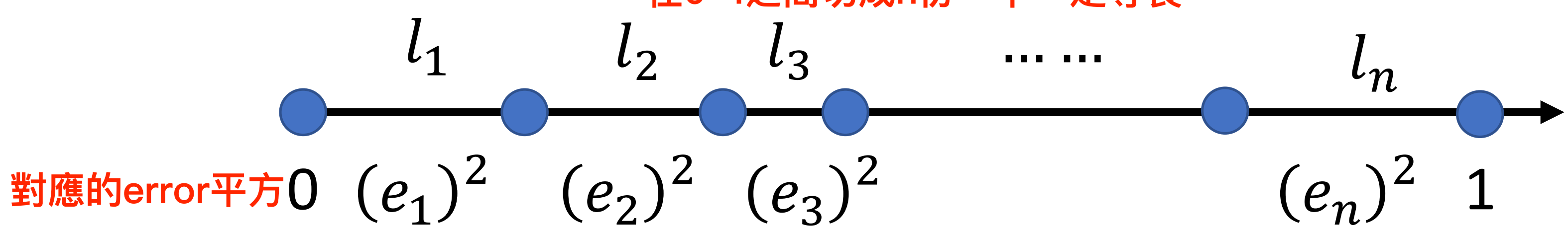
Best of Shallow

The minimum value of e² is $\frac{l^5}{180}$

- If you have n pieces, what is the best way to arrange the n pieces.

$$\sum_{i=1}^{n} l_i = 1$$

在0–1之間切成n份，不一定等長



$$E^2 = \sum_{i=1}^{n} (e_i)^2 = \sum_{i=1}^{n} \frac{(l_i)^5}{180}$$

$$l_i = 1/n$$

The best way is "equal segment"

$$E^2 = \sum_{i=1}^{n} \frac{(1/n)^5}{180} = \frac{1}{180} \frac{1}{n^4}$$

Warning of Math

Hölder's inequality

$$\sum_{i=1}^{n} l_i = 1$$

Minimize $\sum_{i=1}^{n} (l_i)^5$

- Given $\{a_1, a_2, \cdots, a_n\}$ and $\{b_1, b_2, \cdots, b_n\}$

$$\sum_{i=1}^{n} |a_i b_i| \le \left(\sum_{i=1}^{n} |a_i|^p\right)^{1/p} \left(\sum_{i=1}^{n} |b_i|^q\right)^{1/q}$$

$$\boxed{\begin{gathered} \frac{1}{p} + \frac{1}{q} = 1 \\ 1 + \frac{p}{q} = p \qquad 1 - p = -\frac{p}{q} \end{gathered}}$$

- Given $\{l_1, l_2, \cdots, l_n\}$ and $\{1, 1, \cdots, 1\}$

$$\underbrace{\boxed{\sum_{i=1}^{n} l_i}}_{=1} \le \left(\sum_{i=1}^{n} {l_i}^p\right)^{1/p} \Bigg(\underbrace{\boxed{\sum_{i=1}^{n} 1^q}}_{=n}\Bigg)^{1/q}$$

$$n^{-1/q} \le \left(\sum_{i=1}^{n} {l_i}^p\right)^{1/p}$$

$$n^{\overset{1-p}{\cancel{-p/q}}} \le \sum_{i=1}^{n} {l_i}^p$$

p=5

$$n^{-4} \le \sum_{i=1}^{n} {l_i}^5$$

End of Warning

Best of Shallow

The minimum value of e^2 is $\frac{l^5}{180}$

- If you have n pieces, what is the best way to arrange the n pieces.

error的lower bound

$$E^2 = \frac{1}{180}\frac{1}{n^4} \Rightarrow E = \sqrt{\frac{1}{180}\frac{1}{n^2}}$$

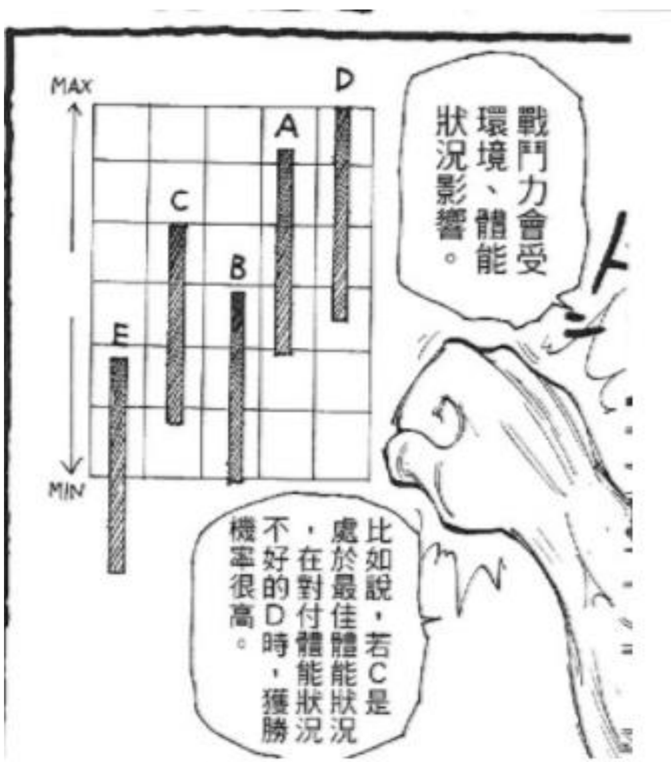
To make $E \le \varepsilon$, what is the n we need?

$$E = \sqrt{\frac{1}{180}\frac{1}{n^2}} \le \varepsilon \qquad n^2 \ge \sqrt{\frac{1}{180}}\frac{1}{\varepsilon} \qquad n \ge \sqrt[4]{\frac{1}{180}}\sqrt{\frac{1}{\varepsilon}}$$

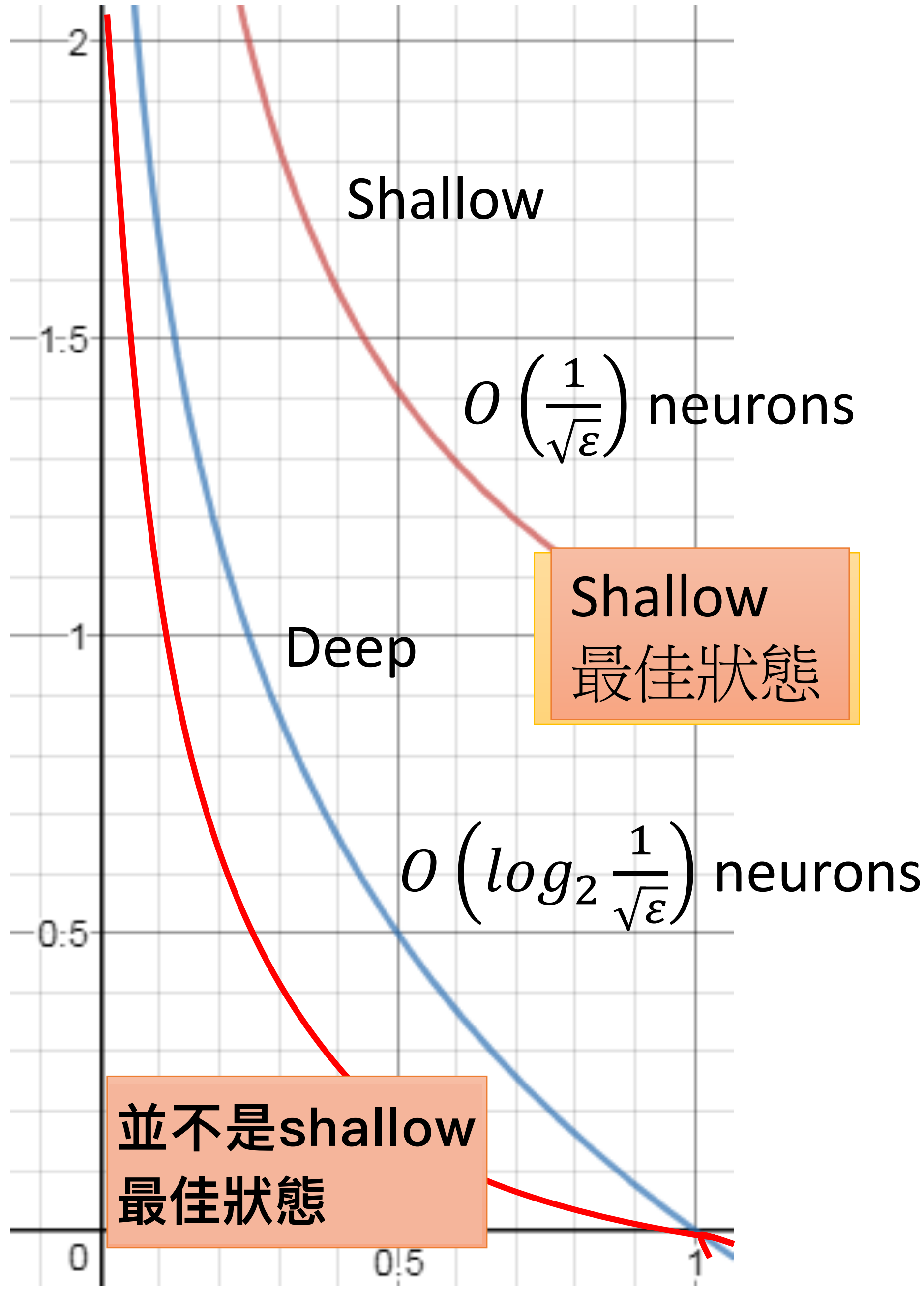
At least $O\left(\frac{1}{\sqrt{\varepsilon}}\right)$ neurons

Deep v.s. Shallow

Deep is exponentially better than shallow.



(獵人第二十卷)



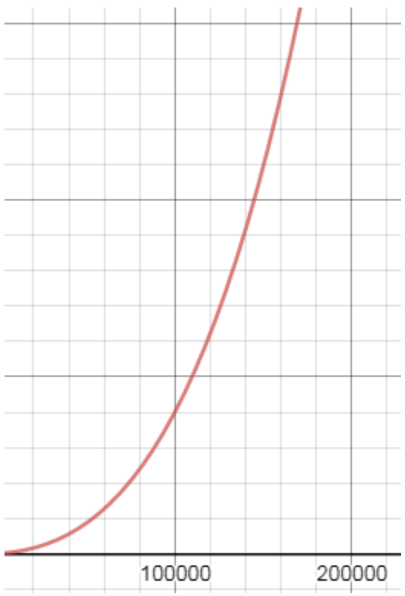
More related theories

More Theories

存在一個function

- A function expressible by a 3-layer feedforward network cannot be approximated by 2-layer network.
  - Unless the width of 2-layer network is VERY large
  - Applied on activation functions beyond relu

不限於relu的activation function



The width of 3-layer network is K.

The width of 2-layer network should be $Ae^{BK^{4/19}}$.

3–layer的neuron數目竟然被放在2–layer的指數部分

Ronen Eldan, Ohad Shamir, "The Power of Depth for Feedforward Neural Networks", COLT, 2016

More Theories

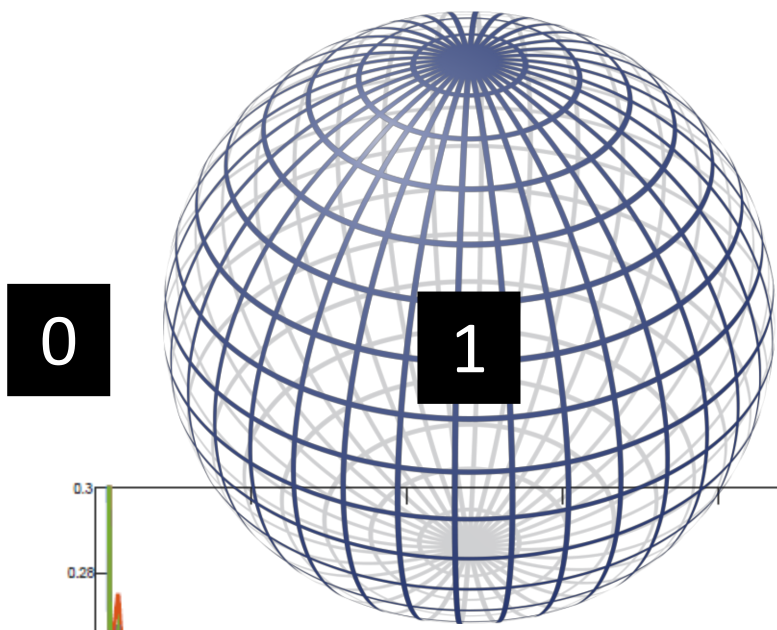
- A function expressible by a deep feedforward network cannot be approximated by a shallow network.
  - Unless the width of the shallow network is VERY large
  - Applied on activation functions beyond relu

Deep Network:

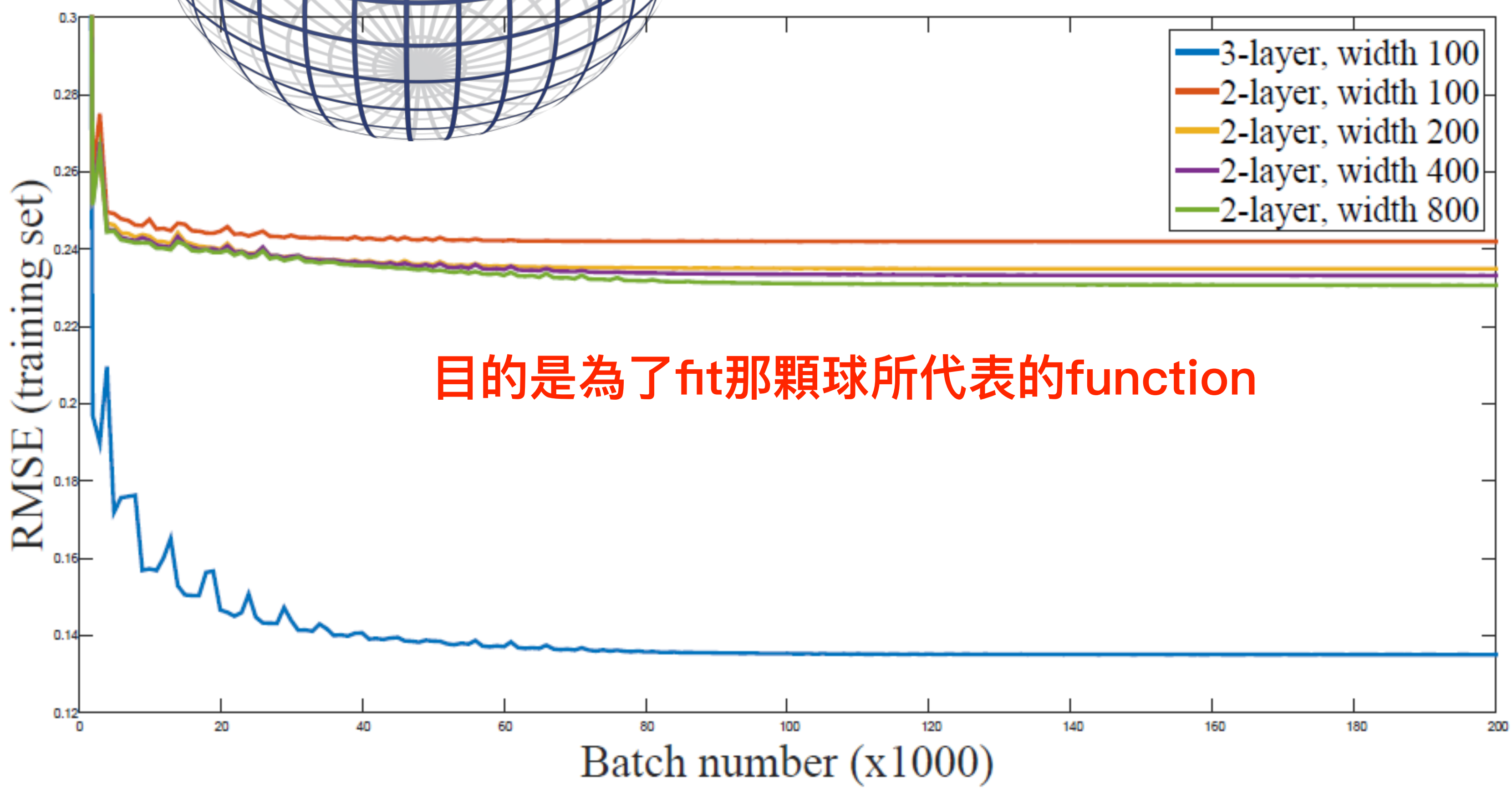
$\Theta(k^3)$ layers, $\Theta(1)$ nodes per layer, $\Theta(1)$ distinct parameters

Shallow Network: $\Theta(k)$ layers $\Rightarrow$ $\Omega\left(2^k\right)$ nodes

Matus Telgarsky, "Benefits of depth in neural networks", COLT, 2016



Itay Safran, Ohad Shamir, “Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks”, ICML, 2017



More Theories

Dmitry Yarotsky, “Error bounds for approximations with deep ReLU networks”, arXiv, 2016
Dmitry Yarotsky, “Optimal approximation of continuous functions by very deep ReLU networks”, arXiv 2018
Shiyu Liang, R. Srikant, “Why Deep Neural Networks for Function Approximation?”, ICLR, 2017
Itay Safran, Ohad Shamir, “Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks”, ICML, 2017

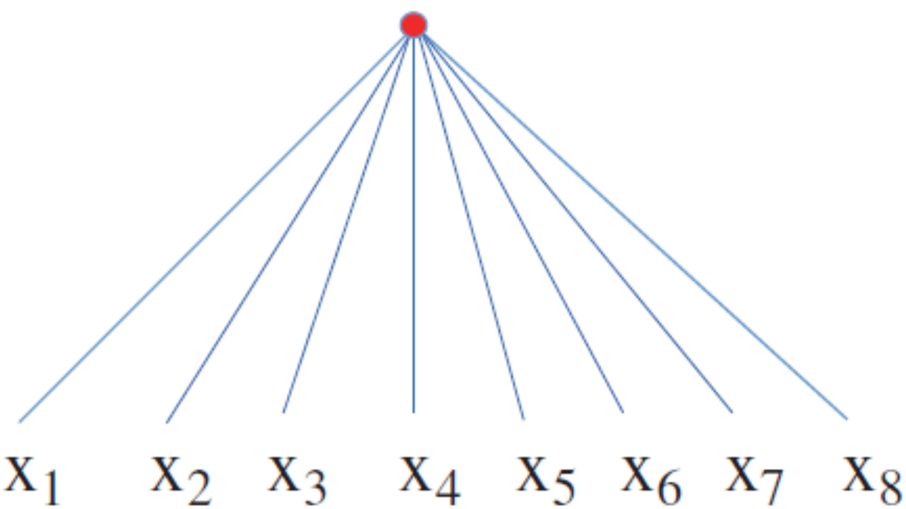
不一定是所有function都滿足deep>shallow，因此每篇paper所假設的function都有一定的複雜度

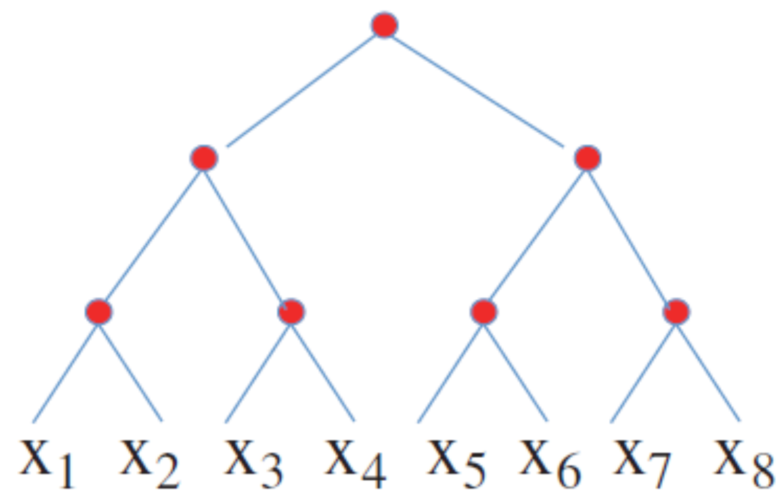
If a function f has “certain degree of complexity”

- Approximating f to accuracy ε in the L2 norm using a fixed depth ReLU network requires at least $poly(1/\varepsilon)$
- There exist a ReLU network of depth and width at most $poly\left(log(1/\varepsilon)\right)$ that can achieve the approximation.

The Nature of Functions

假設objective function是composition structure，則deep>shallow





Hrushikesh Mhaskar, Qianli Liao, Tomaso Poggio, When and Why Are Deep Networks Better Than Shallow Ones?, AAAI, 2017

Concluding Remarks

如果要考慮的function比f(x)=x平方 來得複雜的話，則滿足deep > shallow

而我們所要考慮的function一定比他複雜，因此deep一定是最棒的