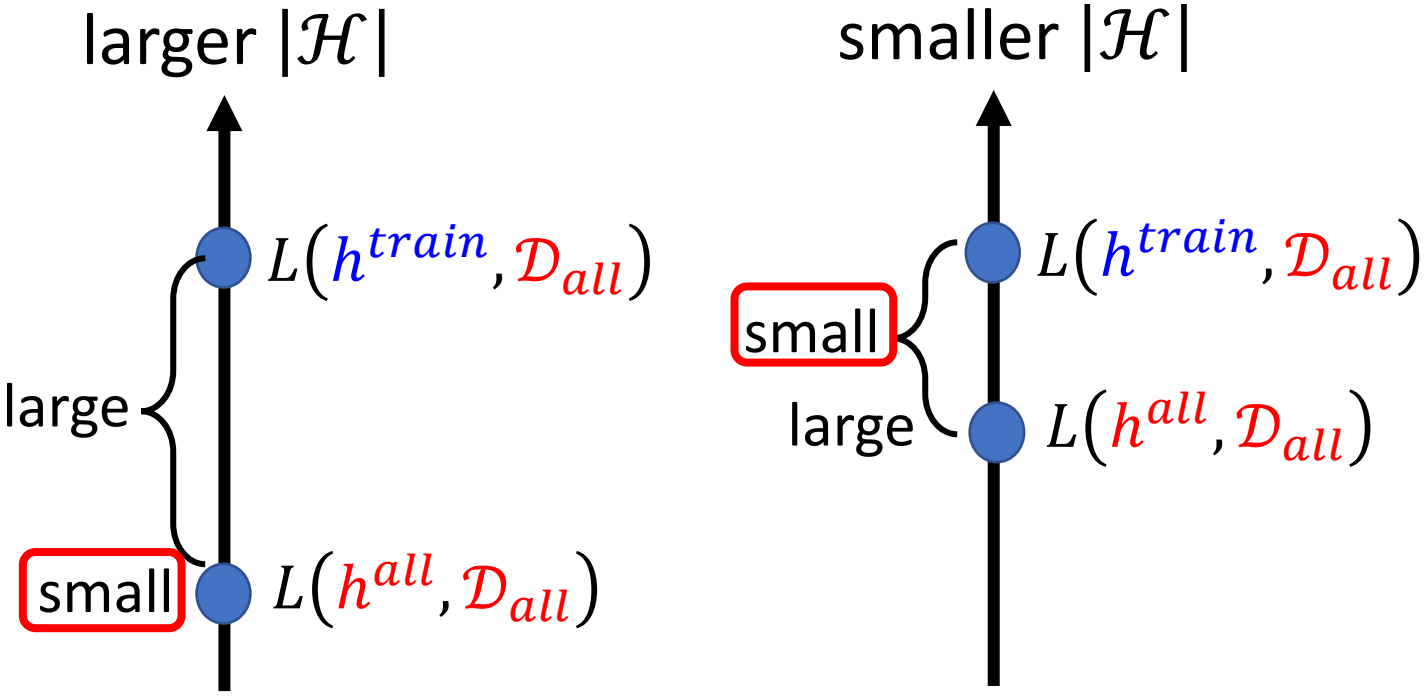
Why Deep Learning?

李宏毅
Hung-yi Lee

魚與熊掌可以兼得嗎？



$$h^{all} = arg \min_{h \in \mathcal{H}} L(h, \mathcal{D}_{all})$$

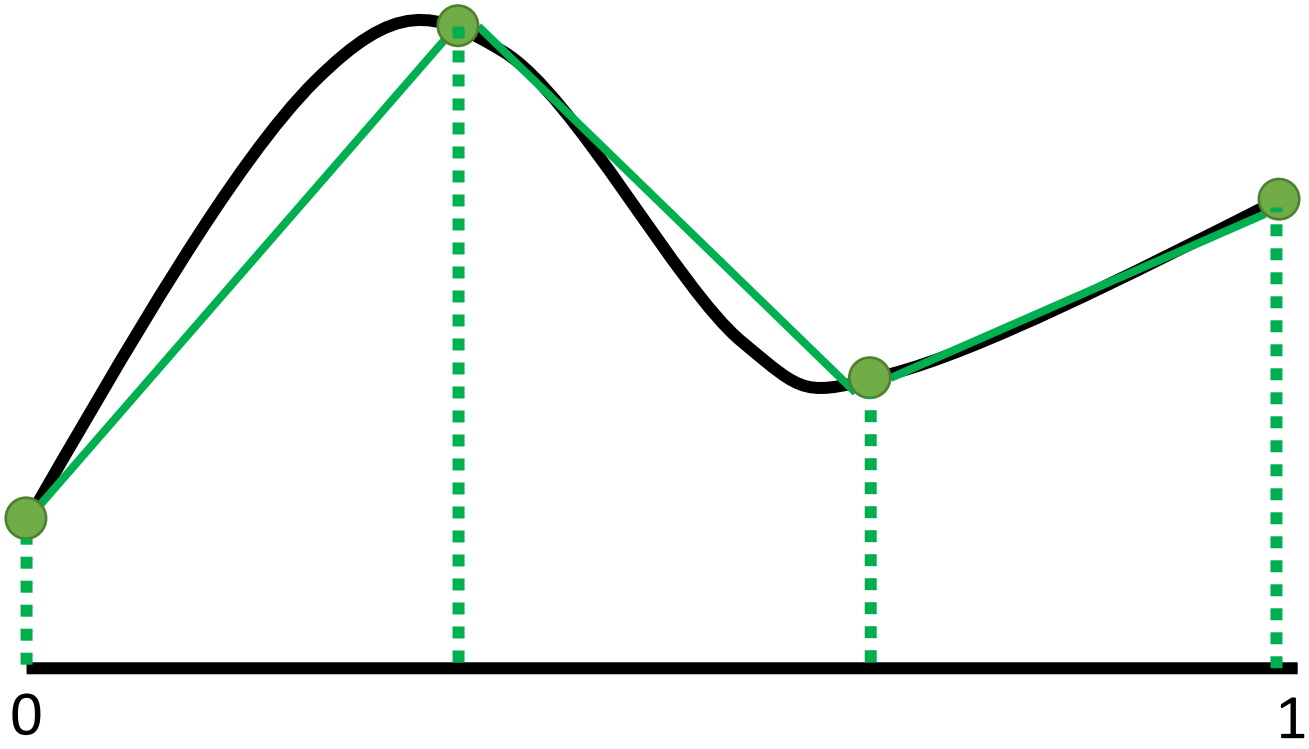
Still low loss

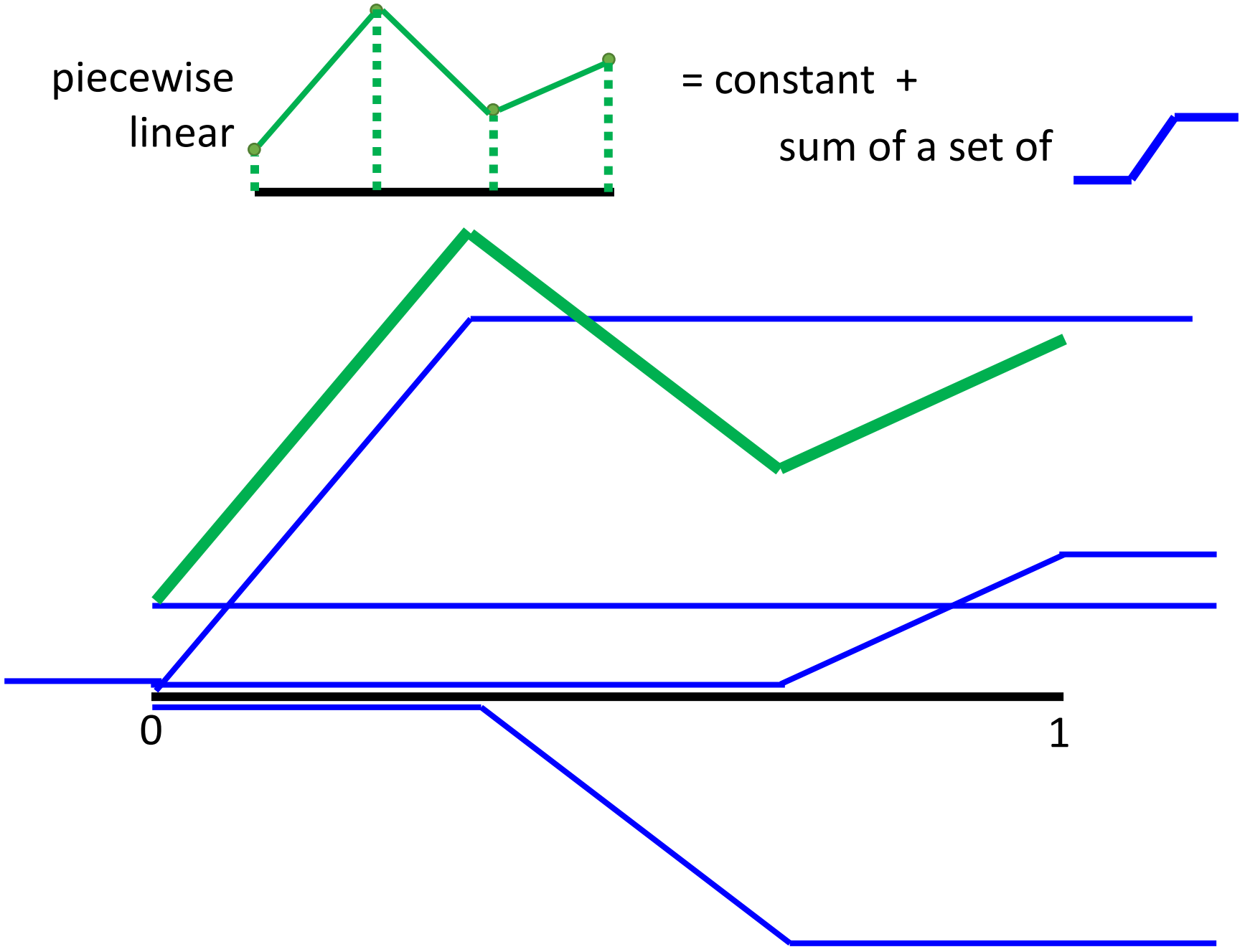
Small (fewer candidates)

Review: Why Hidden Layer?

Piecewise Linear

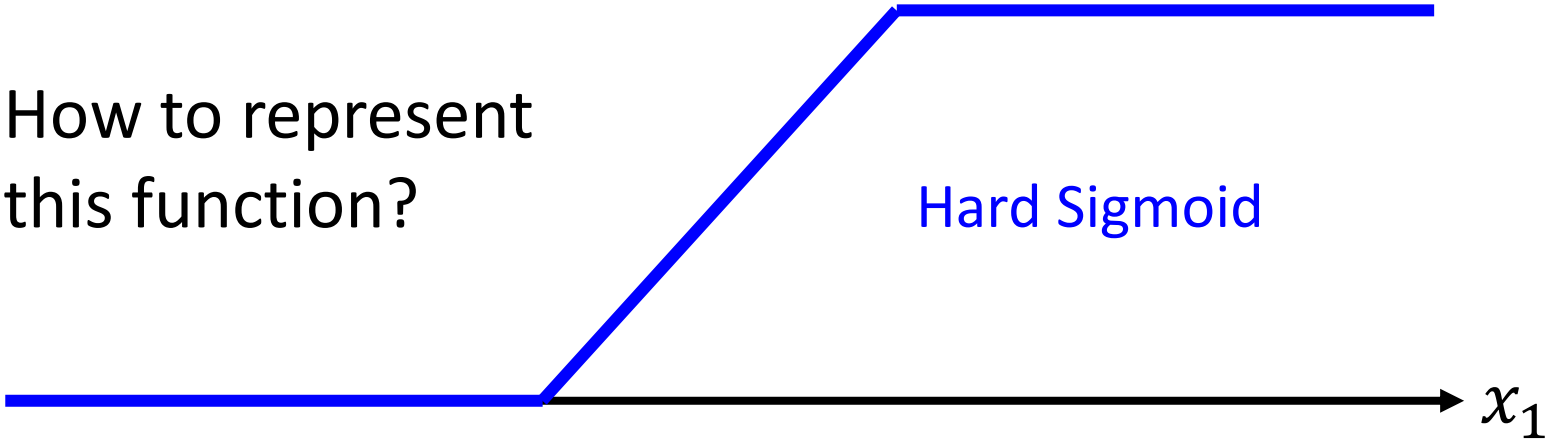
We can have good approximation with sufficient pieces.




piecewise
linear
= constant +
sum of a set of
0
1

Piecewise linear = constant + sum of a set of

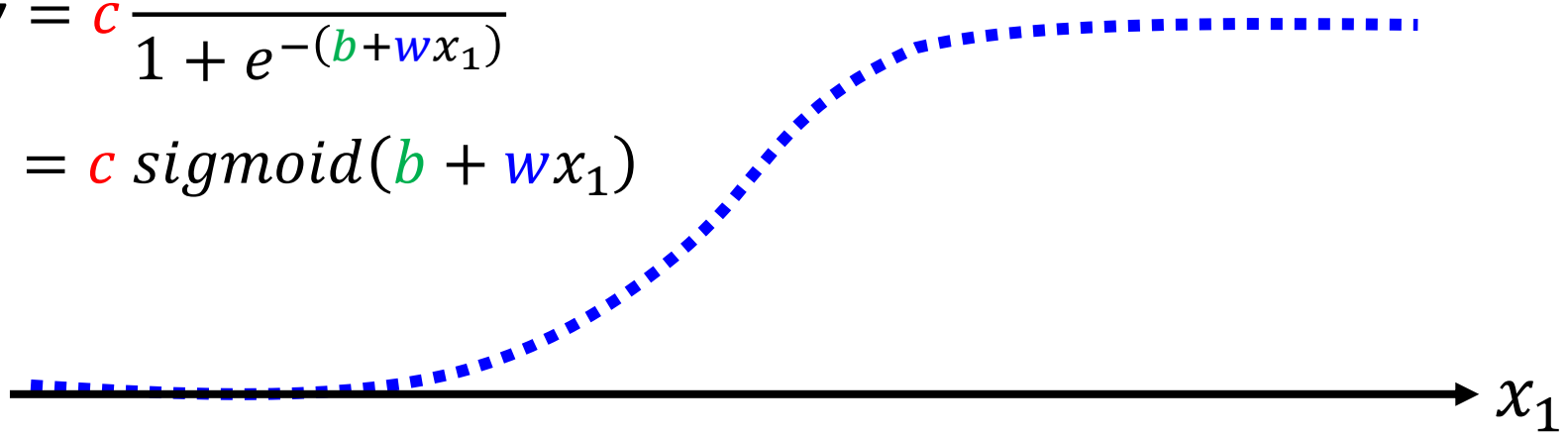
How to represent this function?



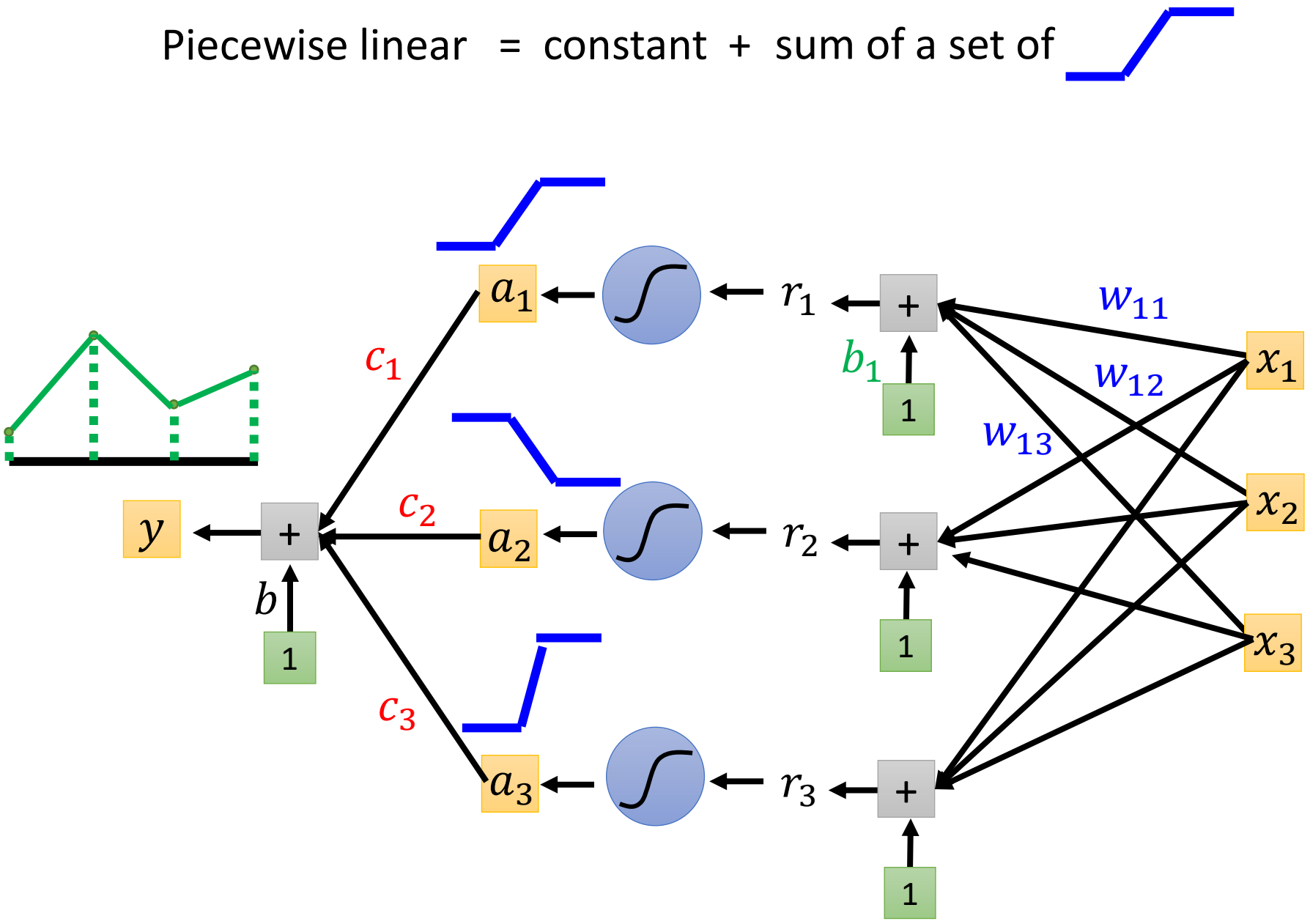
Sigmoid Function

$$y = c\frac{1}{1+e^{-(b+wx_1)}}$$

$$= c\ sigmoid(b + wx_1)$$

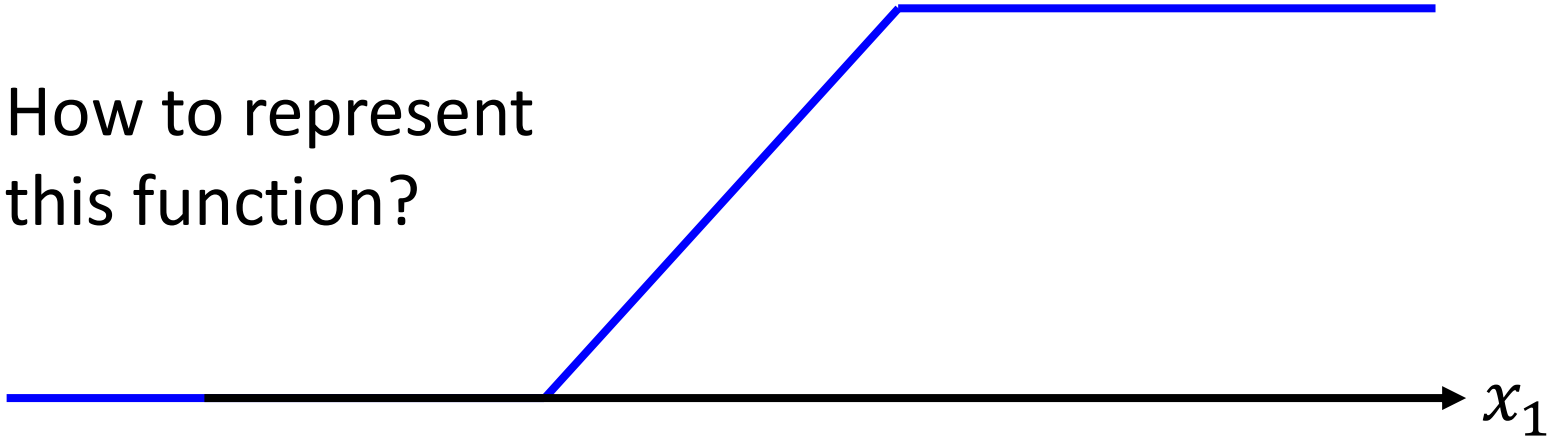


Piecewise linear = constant + sum of a set of

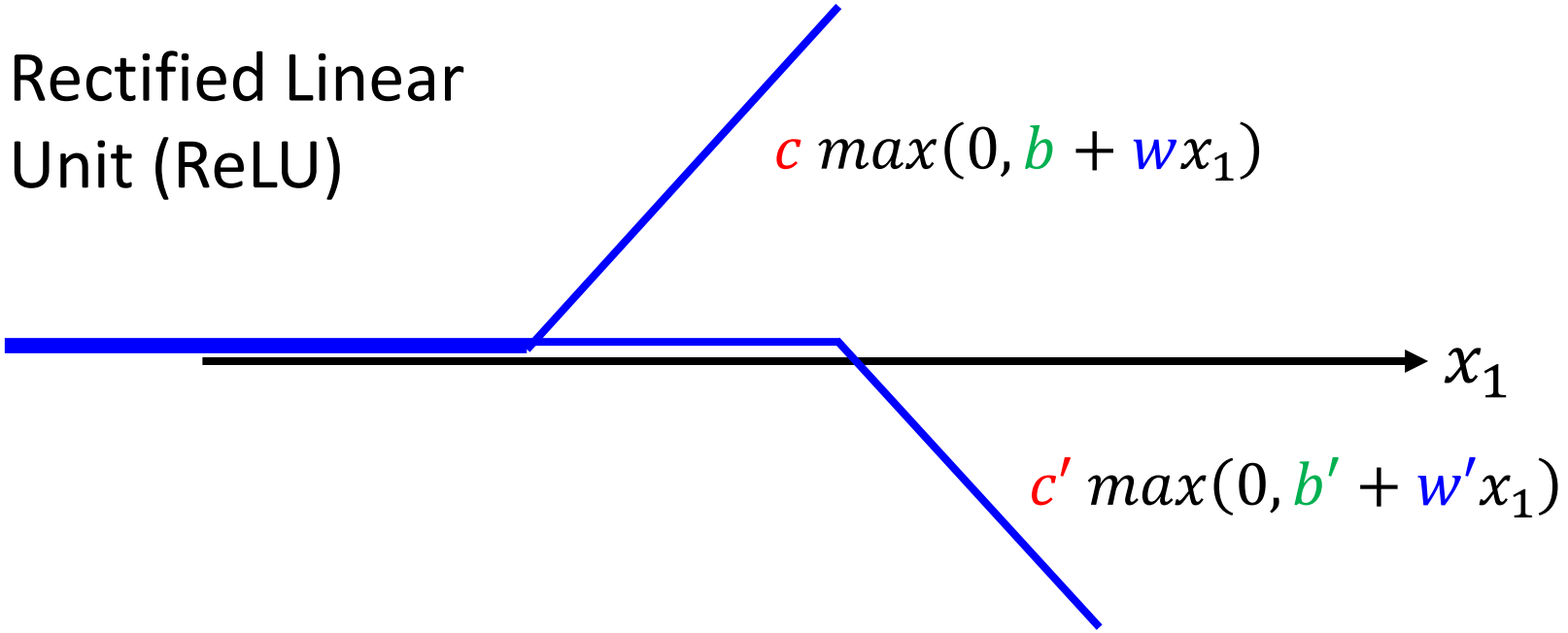


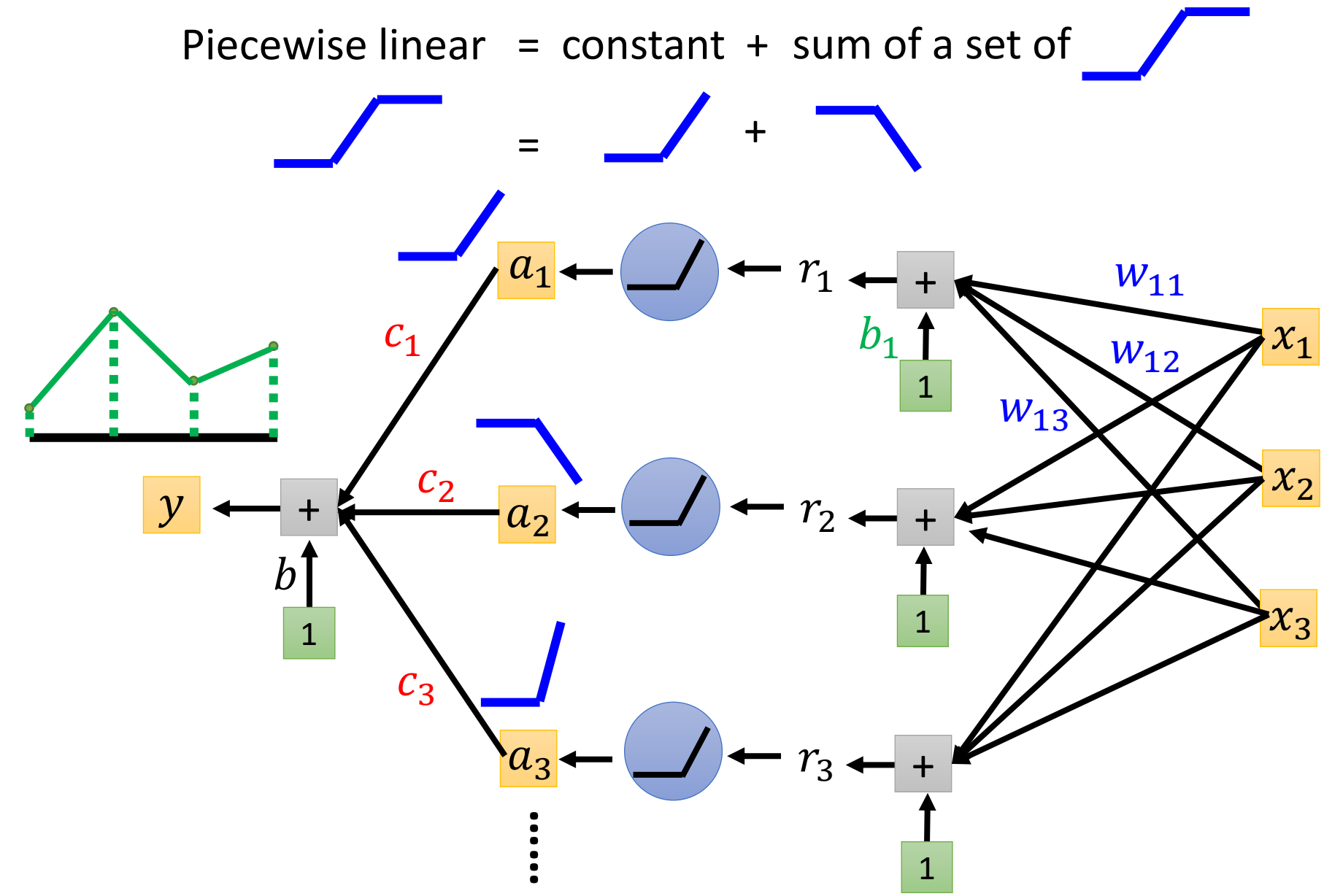
Hard Sigmoid → ReLU

How to represent this function?



Rectified Linear Unit (ReLU)





Why we want "***Deep***" network, not "***Fat***" network?

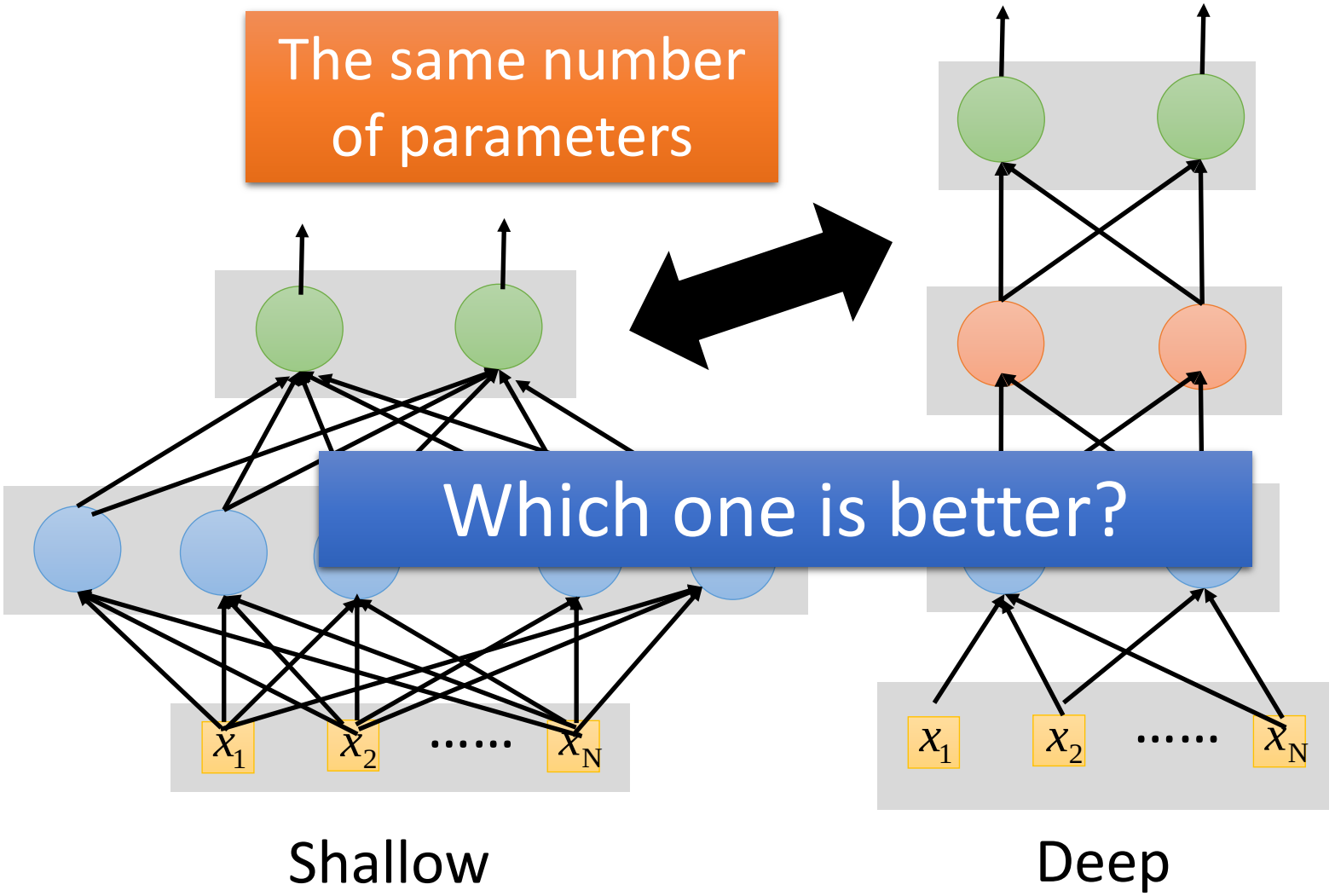
Deeper is Better?

Layer X Size	Word Error Rate (%)
1 X 2k	24.2
2 X 2k	20.4
3 X 2k	18.4
4 X 2k	17.8
5 X 2k	17.2
7 X 2k	17.1

Not surprised, more parameters, better performance

Seide Frank, Gang Li, and Dong Yu. "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." *Interspeech*. 2011.

Fat + Short v.s. Thin + Tall



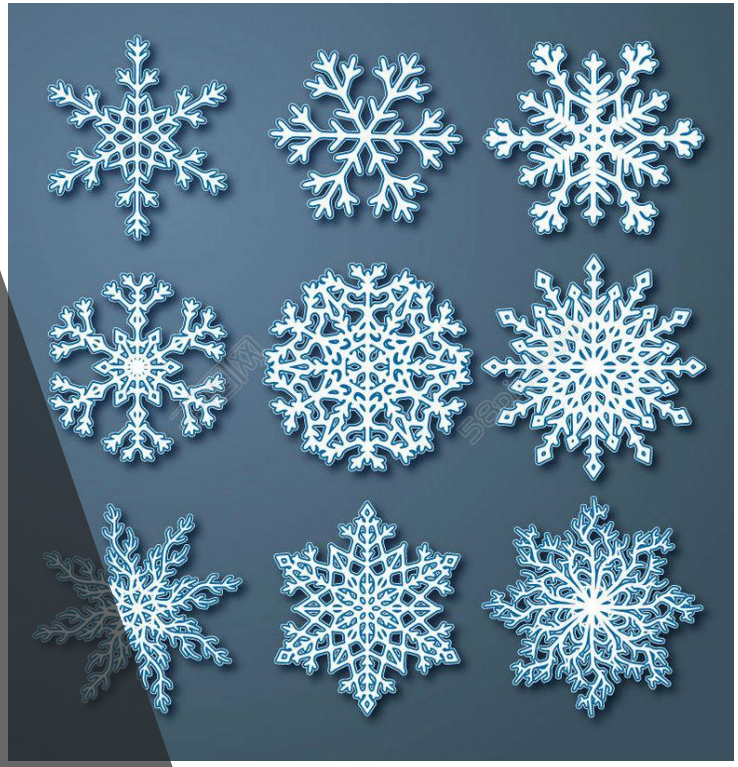
Fat + Short v.s. Thin + Tall

Layer X Size	Word Error Rate (%)	Layer X Size	Word Error Rate (%)
1 X 2k	24.2		
2 X 2k	20.4		
3 X 2k	18.4		
4 X 2k	17.8		
5 X 2k	17.2	1 X 3772	22.5
7 X 2k	17.1	1 X 4634	22.6
		1 X 16k	22.1

Why?

Seide Frank, Gang Li, and Dong Yu. "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." *Interspeech*. 2011.

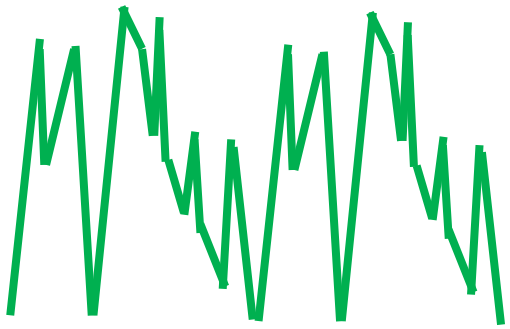
Why we need deep?

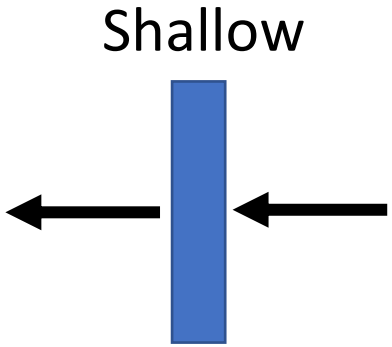


Yes, one hidden layer can represent any function.

However, using deep structure is more effective.

Why we need deep?

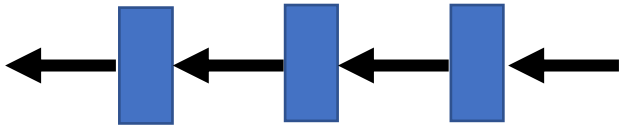




Shallow

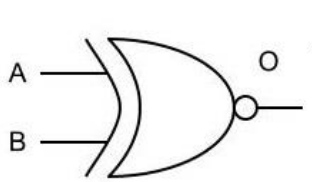
v

More parameters



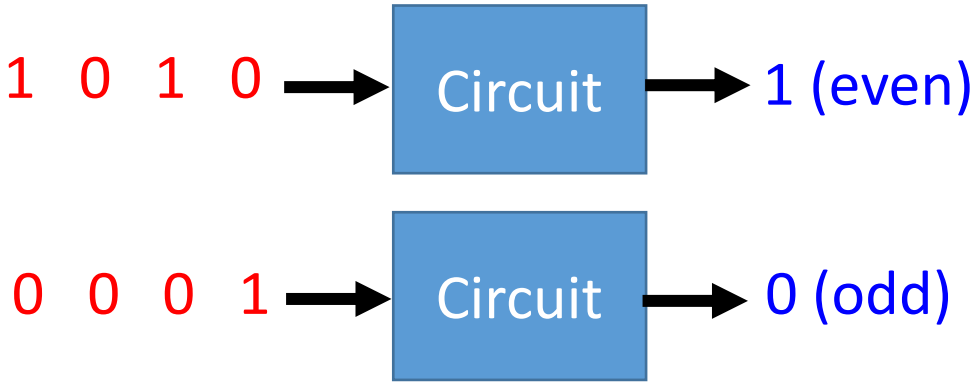
Deep

Analogy – Logic Circuits



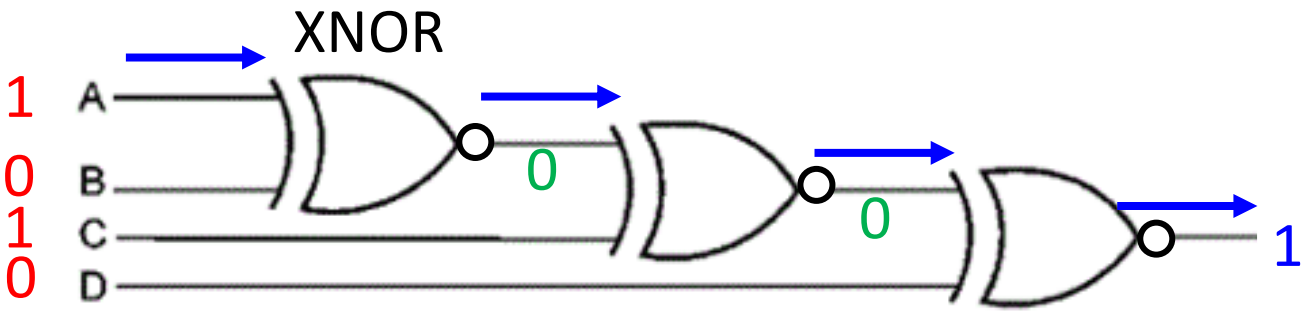
A	B	O
0	0	1
0	1	0
1	0	0
1	1	1

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



For input sequence with d bits,

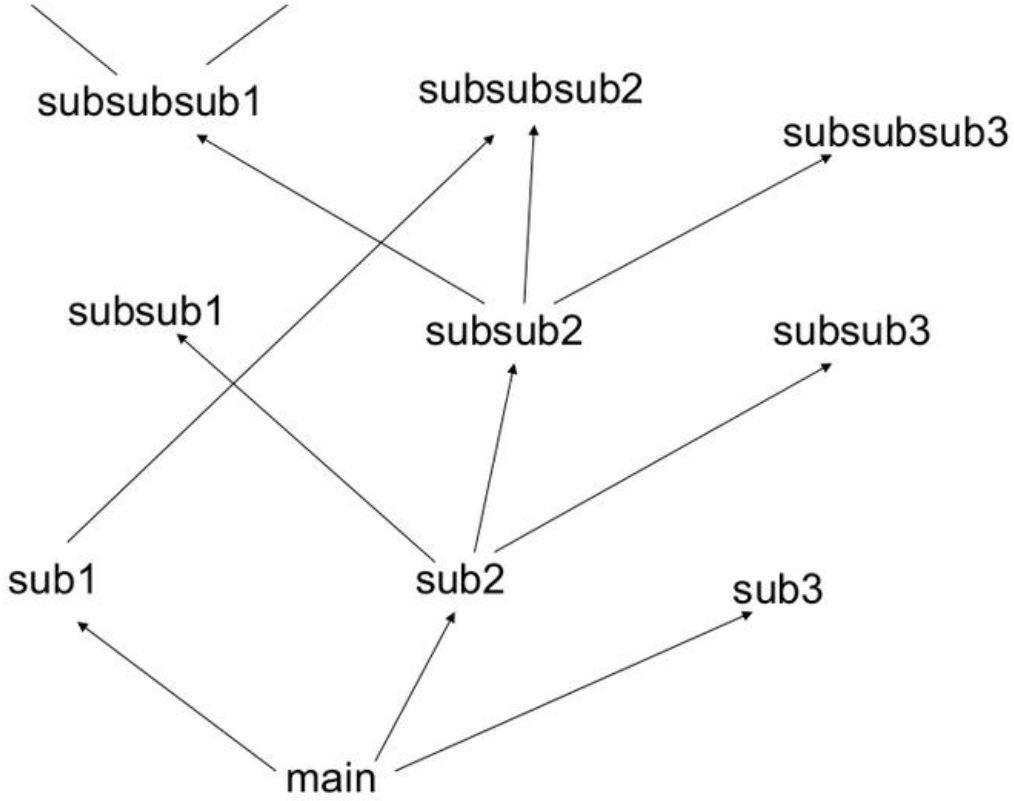
Two-layer circuit need $O(2^d)$ gates.



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

Analogy – Programming

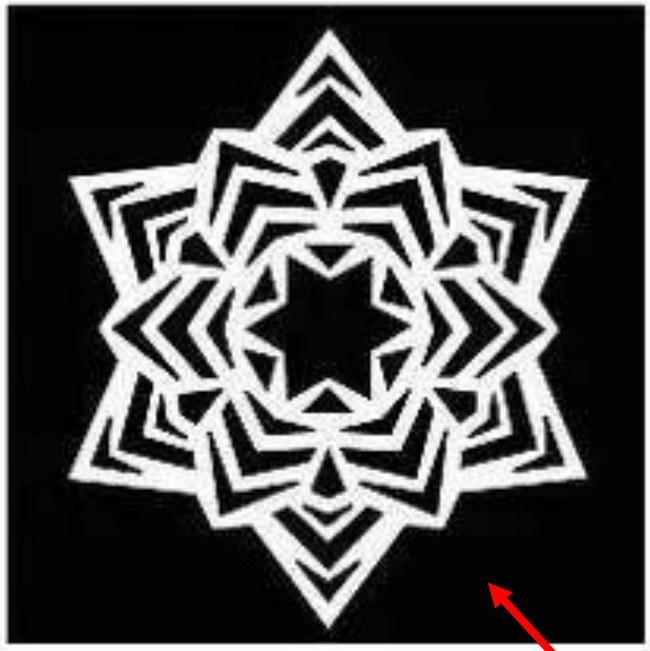
Don’t put everything in your main function.

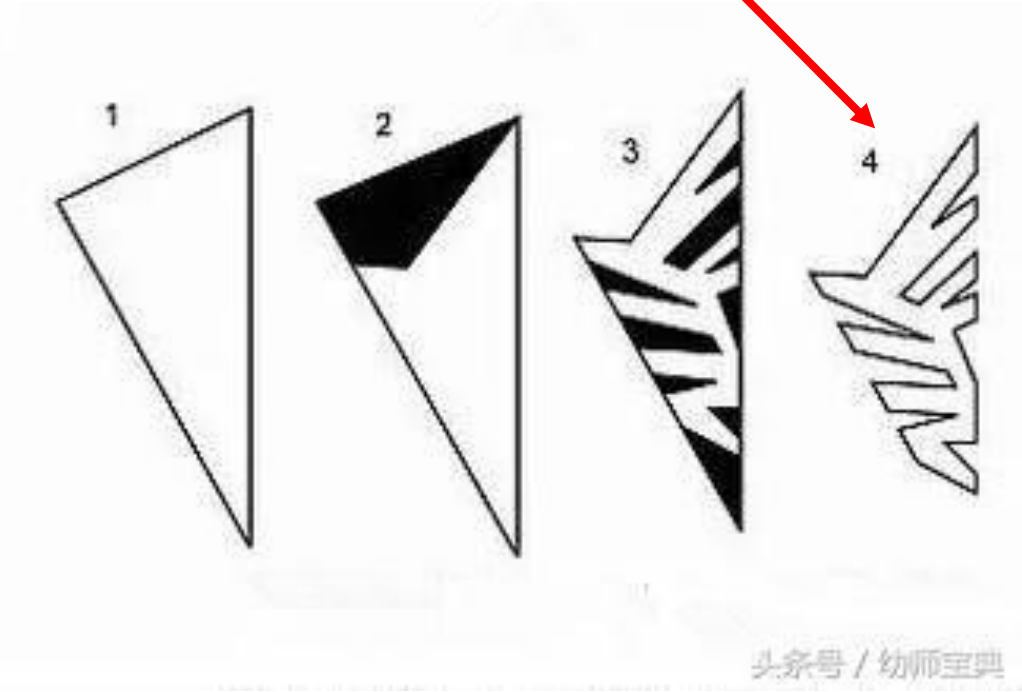


http://rinuboney.github.io/2015/10/18/theoretical-motivations-deep-learning.html

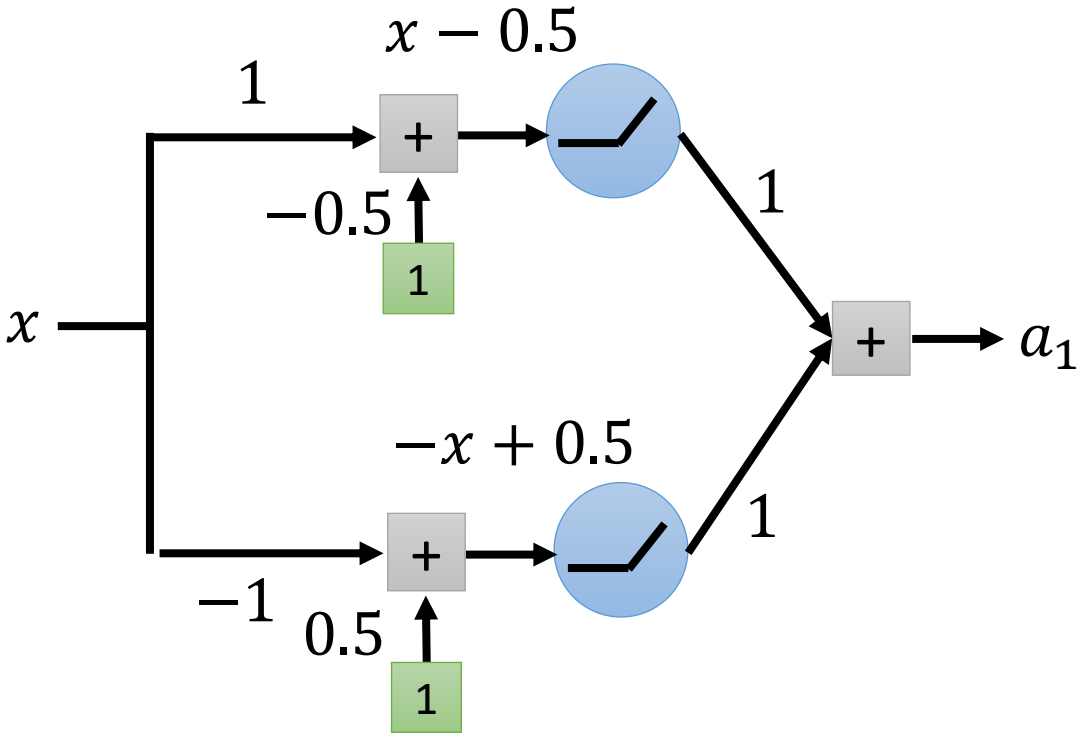
More Analogy

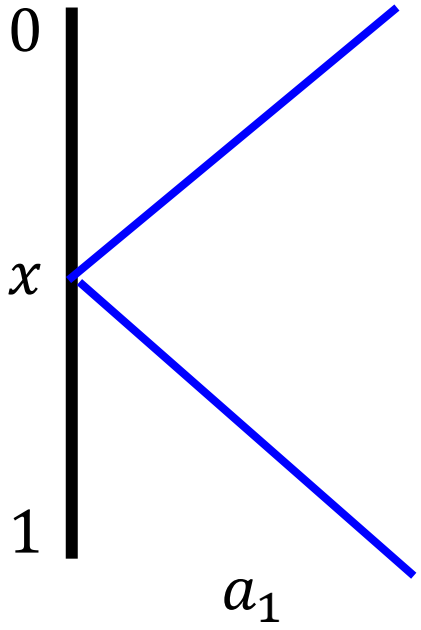
比較有效率

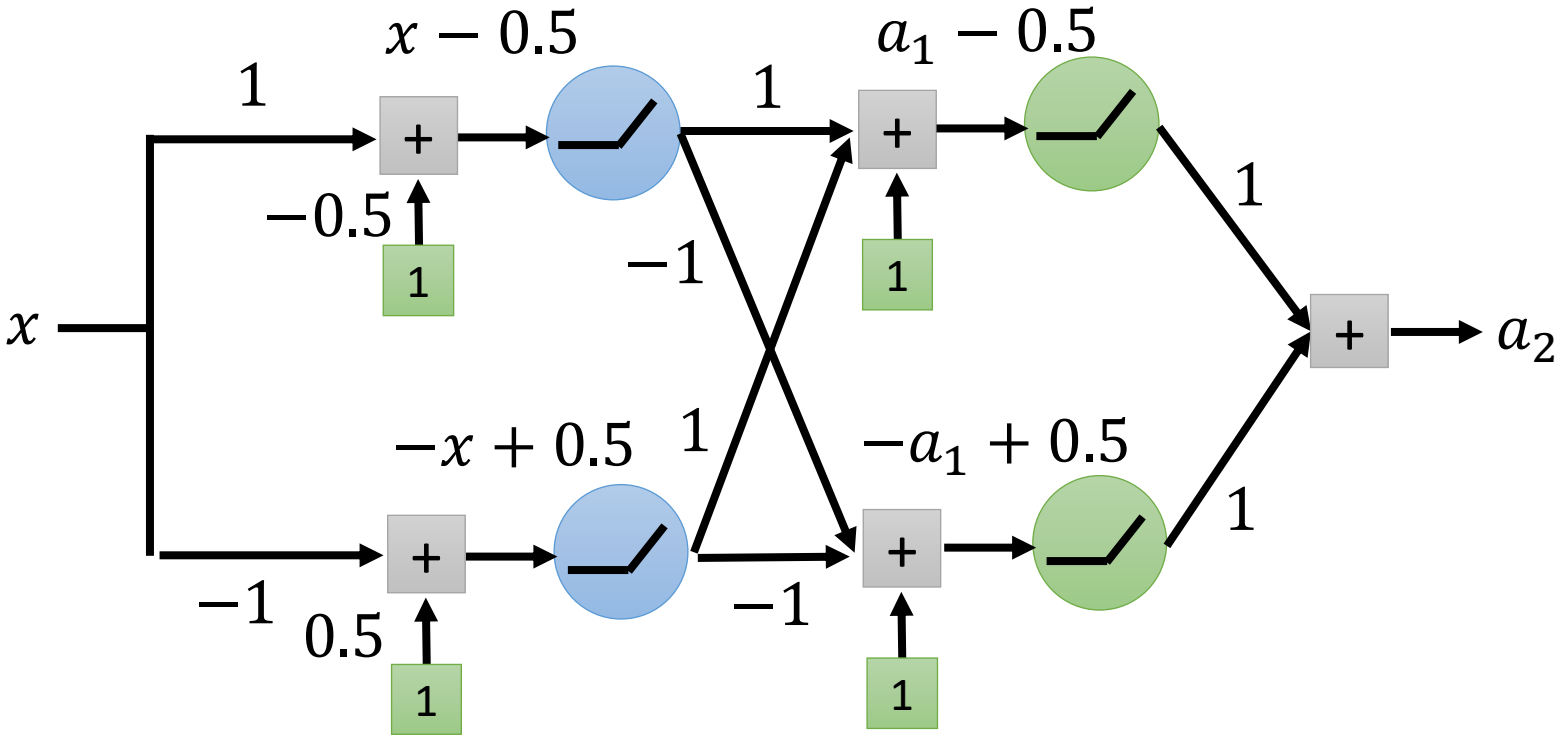


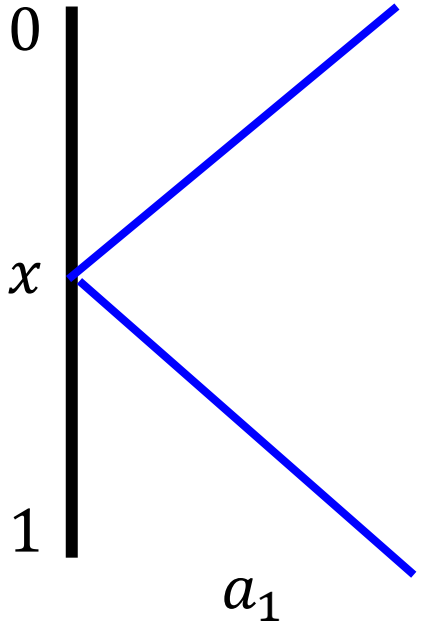


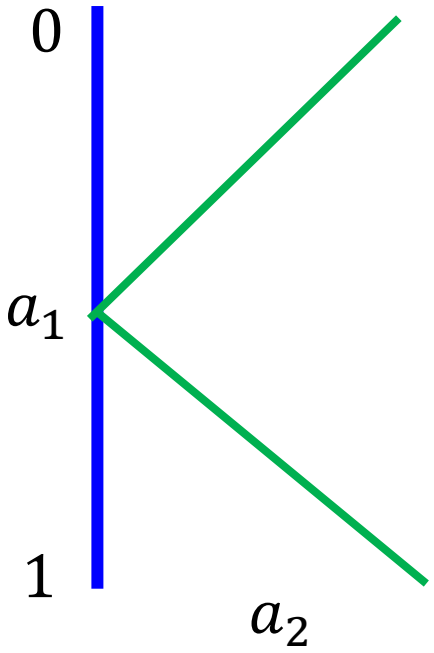
剪很多刀

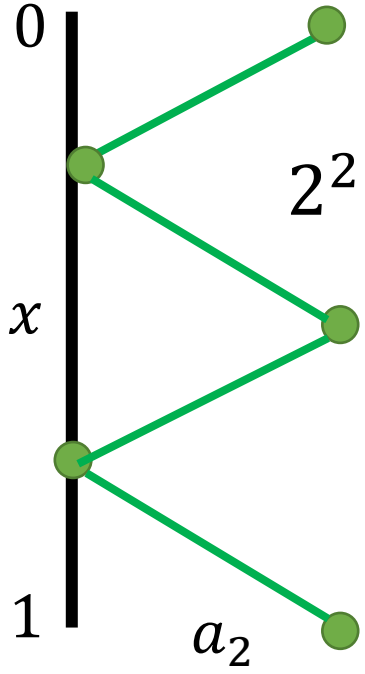

$x - 0.5$
1
+
-0.5
1
x
1
+
a_1
$-x + 0.5$
+
-1
0.5
1
1


0
x
1
a_1

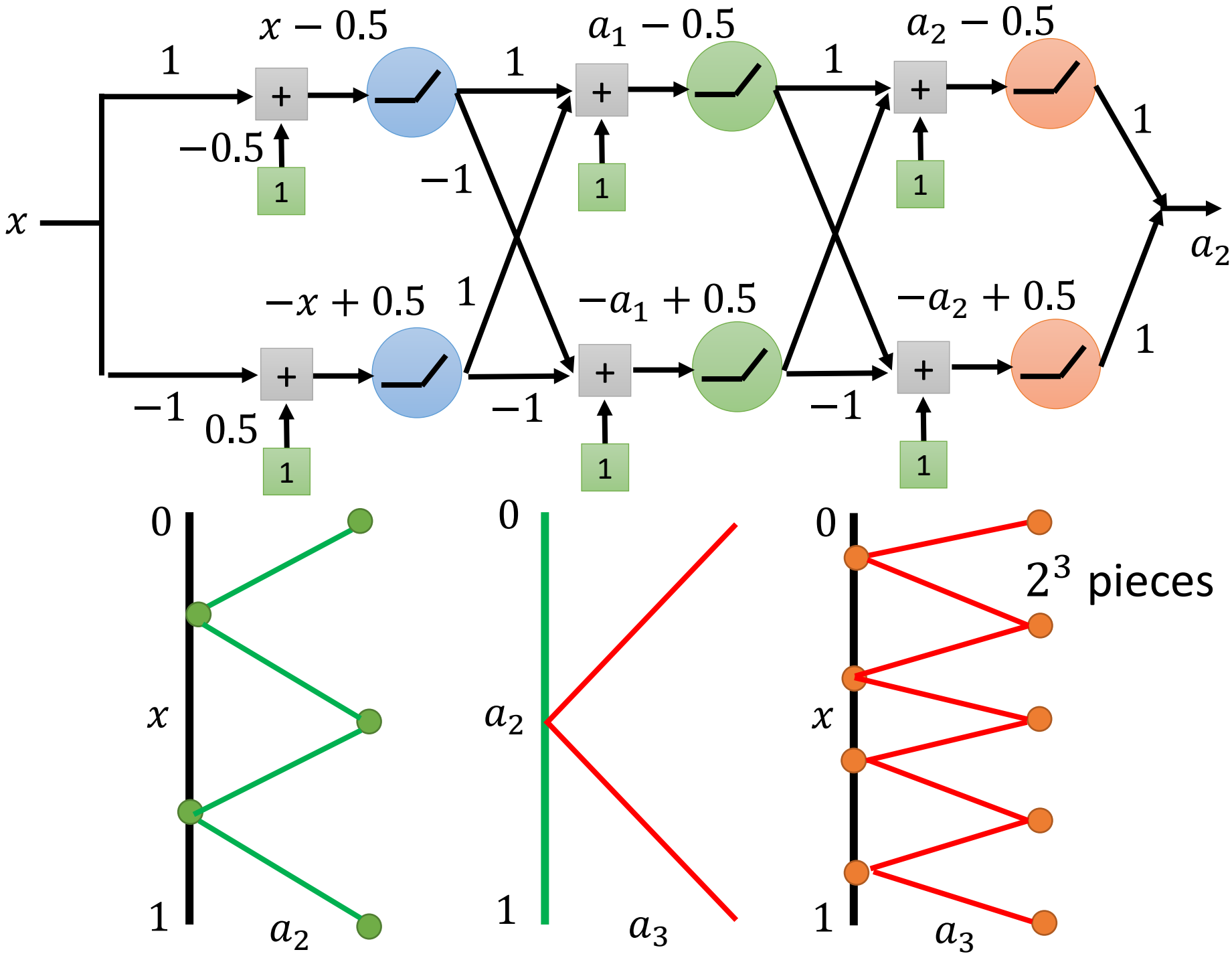




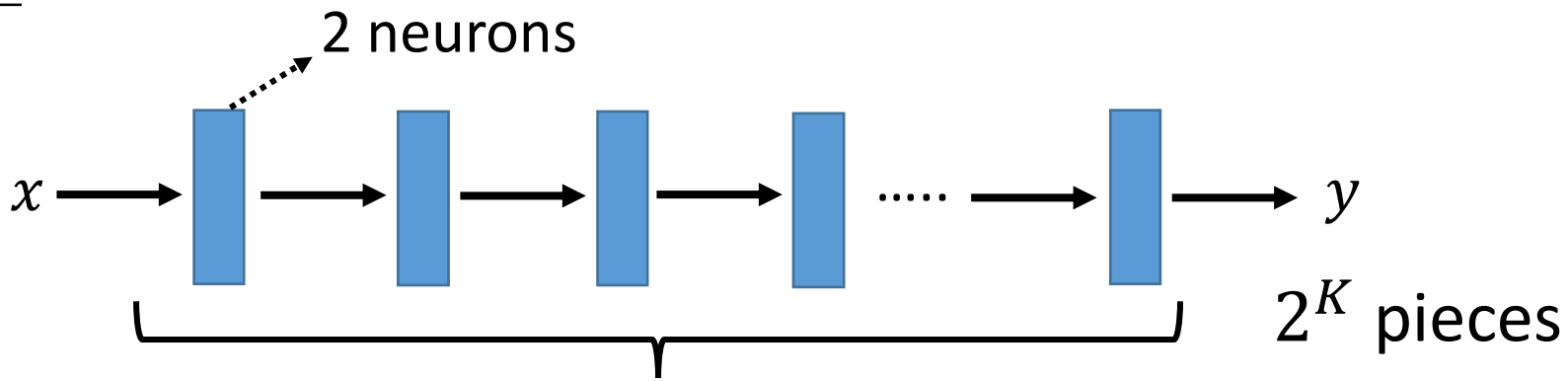




2^2 pieces

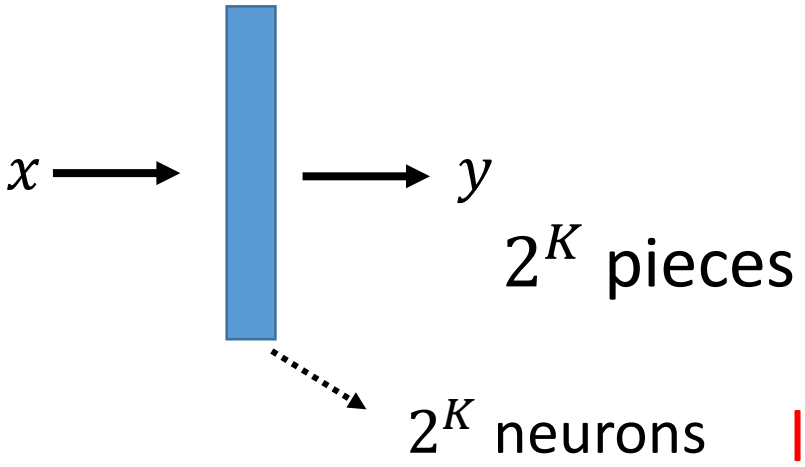

x
$x - 0.5$
$-x + 0.5$
$a_1 - 0.5$
$-a_1 + 0.5$
$a_2 - 0.5$
$-a_2 + 0.5$
a_2
1
-0.5
0.5
-1
0
x
1
a_2
a_2
1
a_3
2^3 pieces
x
1
a_3

Deep

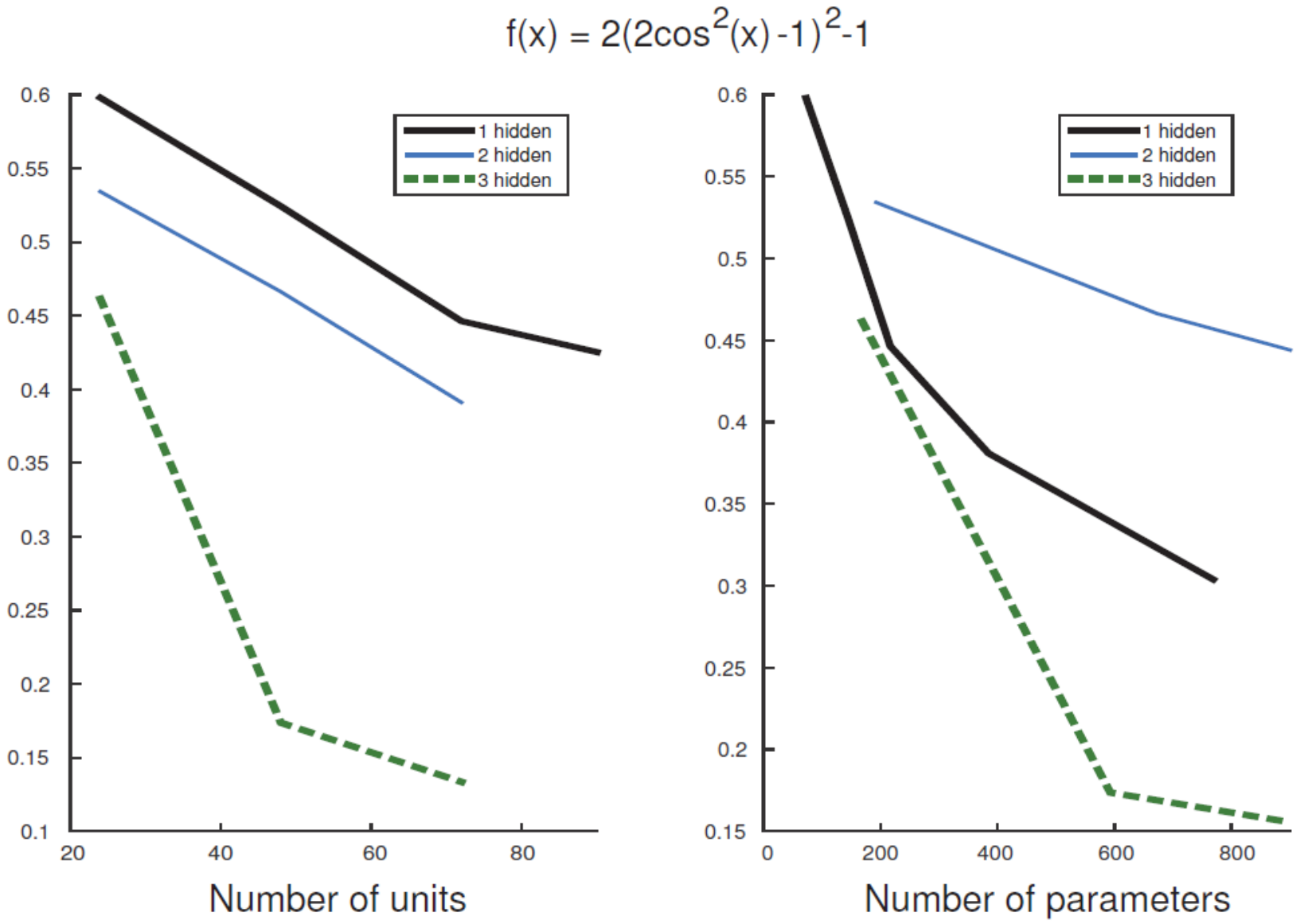


smaller $|\mathcal{H}|$

Shallow



larger $|\mathcal{H}|$



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

Thinks more ……

- Deep networks outperforms shallow ones when the required functions are complex and regular.

  Image, speech, etc. have this characteristics.

- Deep is exponentially better than shallow even when $y = x^2$.

https://youtu.be/FN8jclCrqY0

https://youtu.be/qpuLxXrHQB4