Number of output regions

output = 1

500 hidden

D_i : # inputs

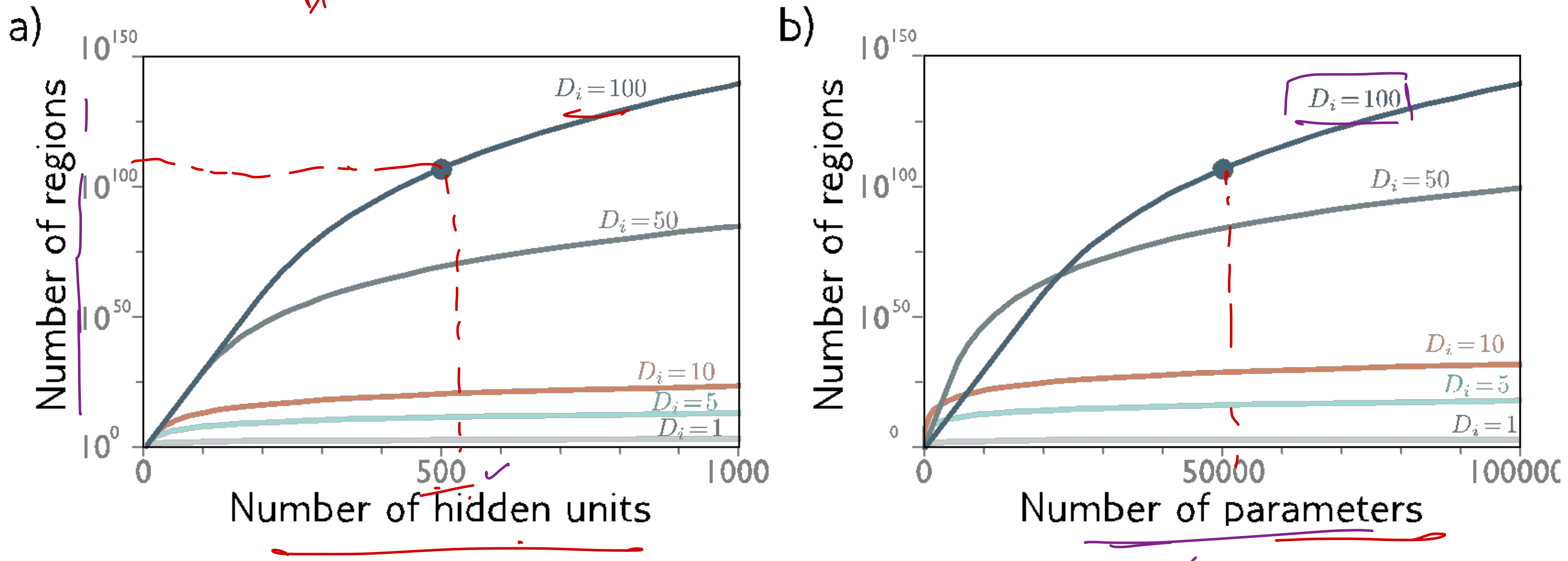
D : # hidden units

- In general, each output consists of D dimensional convex polytopes

100 i/p 1 o/p

- How many?

D_i : i/p dim



Highlighted point = 500 hidden units or 51,001 parameters

Number of regions created by $D > D_i$ planes in D_i-dimensions was proved to be

i/p dim

hidden units

$$N = \sum_{j=0}^{D_i} \binom{D}{j}$$

[Binomial coefficients]

$$\binom{D}{0} + \binom{D}{1} + \cdots \binom{D}{D_i}$$

If $D = D_i$ $\qquad 2^{D_i} = 2^{D}$

$$\boxed{2^{D_i} < N < 2^{D}}$$

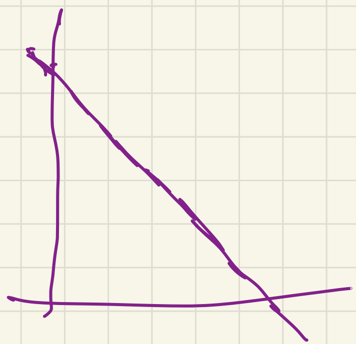
$D_i = 1$

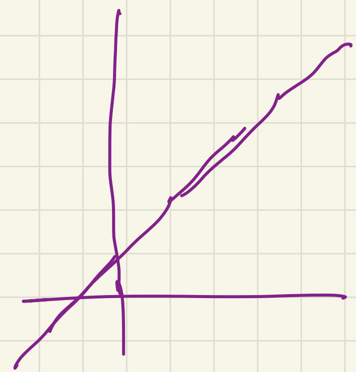
[D + 1 regions]

$$\sum_{j=0}^{D_i} \binom{D}{j} = \binom{D}{0} + \binom{D}{1} = 1 + D$$

$D_i = 2$

$D = 2$





→ 4 regions

$$\binom{2}{0} + \binom{2}{1} + \binom{2}{2}$$

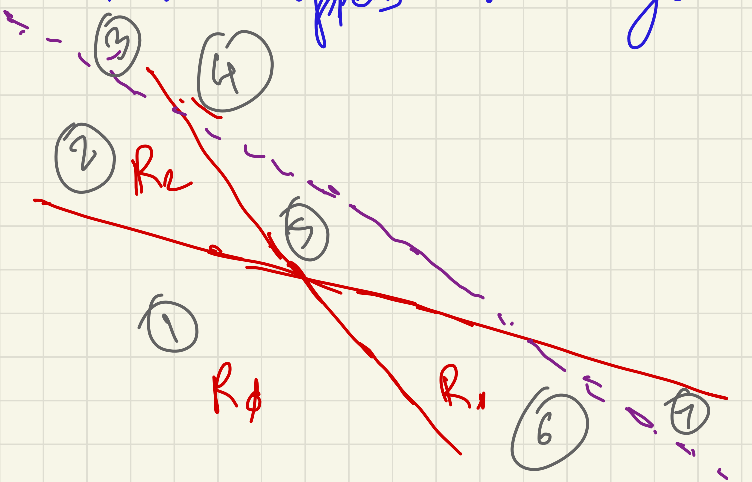
$$= 1 + 2 + 1 = 4$$

$D_i = 2 \Rightarrow \binom{D}{0} + \binom{D}{1} + \binom{D}{2} = 1 + D + \frac{D(D-1)}{2}$

Proof :⇒ Induction on D

Assume true for D $\quad 1 + D + \frac{D(D-1)}{2}$ regions

What happens when you add a new region ?



$(D+1)$th plane can cut all the D planes

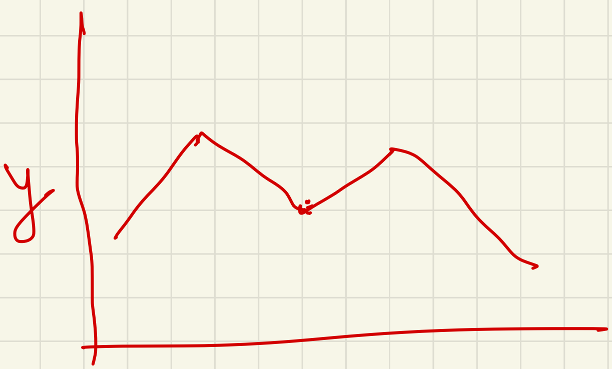
$\Rightarrow$ $D+1$ new regions

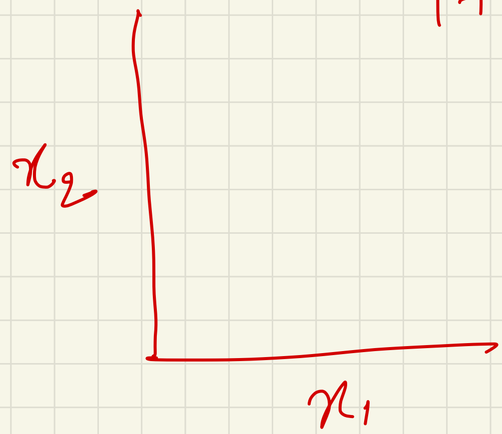
$$1 + D + \frac{D(D-1)}{2} + D + 1$$

$$= 1 + (D+1) + \frac{D(D+1)}{2} \rightarrow \text{proof completed}$$

$$1 + 3 + \frac{3 \cdot 2}{2} = 7$$

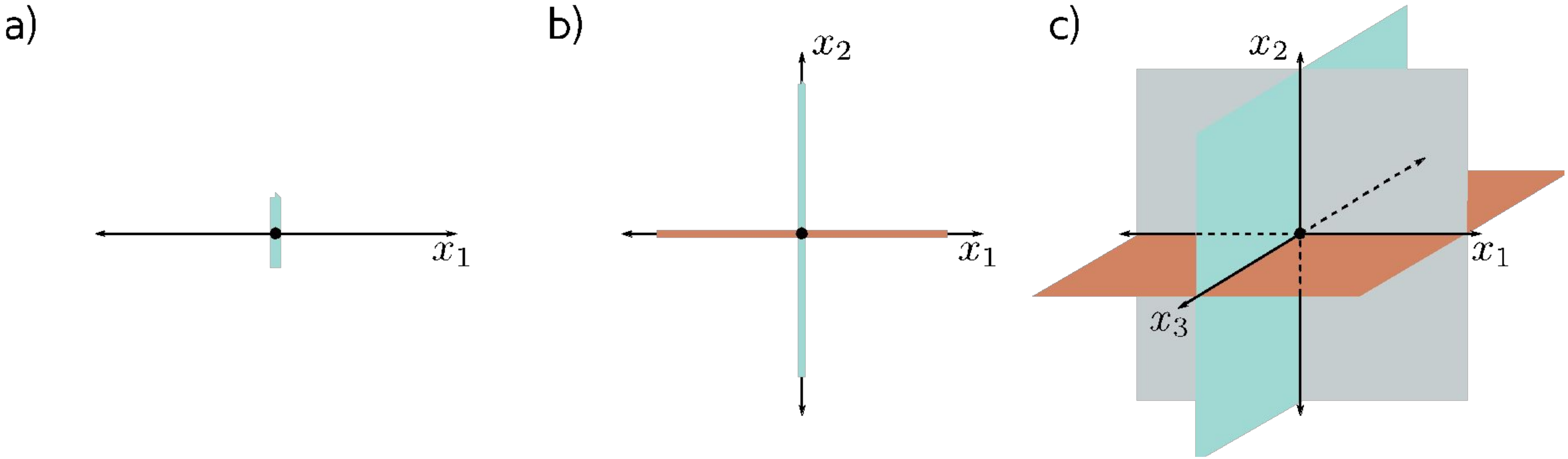
$$1 + 2x_1 + 3x_2$$





$101 \times 500 + 501 \times 1$

$D_i = 100 \qquad D = 500 \qquad \boxed{o/p = 1}$



1D input with 1 hidden unit creates two regions (one joint)

2D input with 2 hidden units creates four regions (two lines)

3D input with 3 hidden units creates eight regions (three planes)

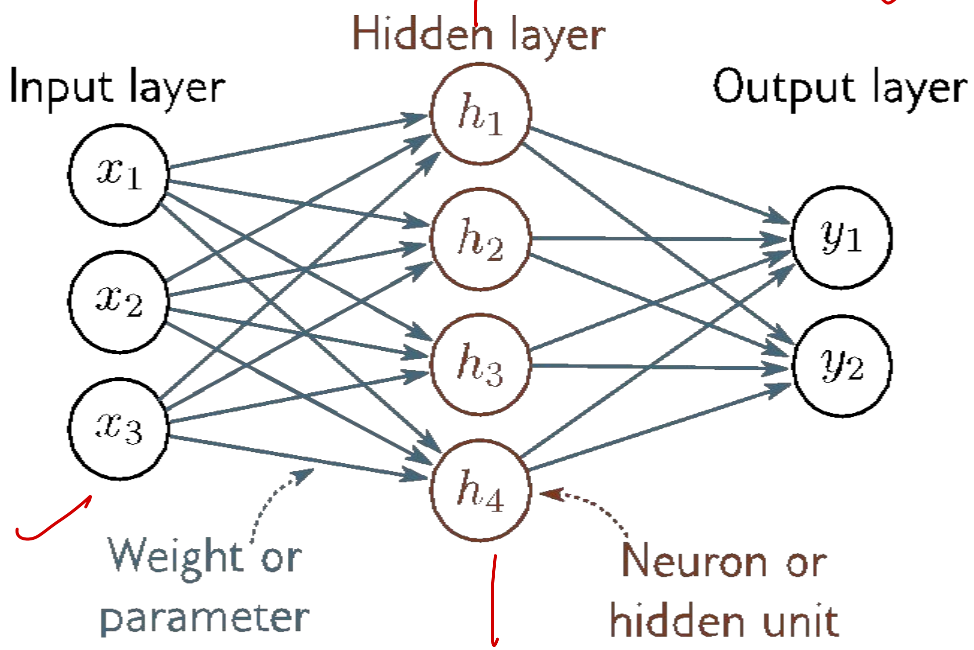
Shallow neural networks

- Example network, 1 input, 1 output
- Universal approximation theorem
- More than one output
- More than one input
- General case
- Number of regions
- Terminology

Nomenclature

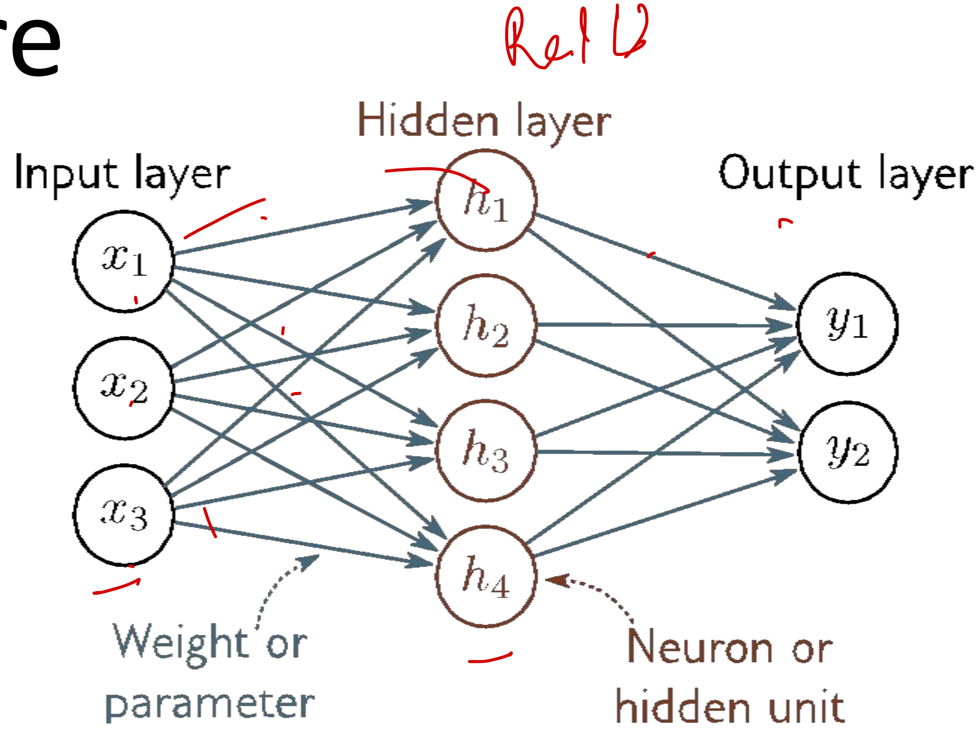
Shallow

NN



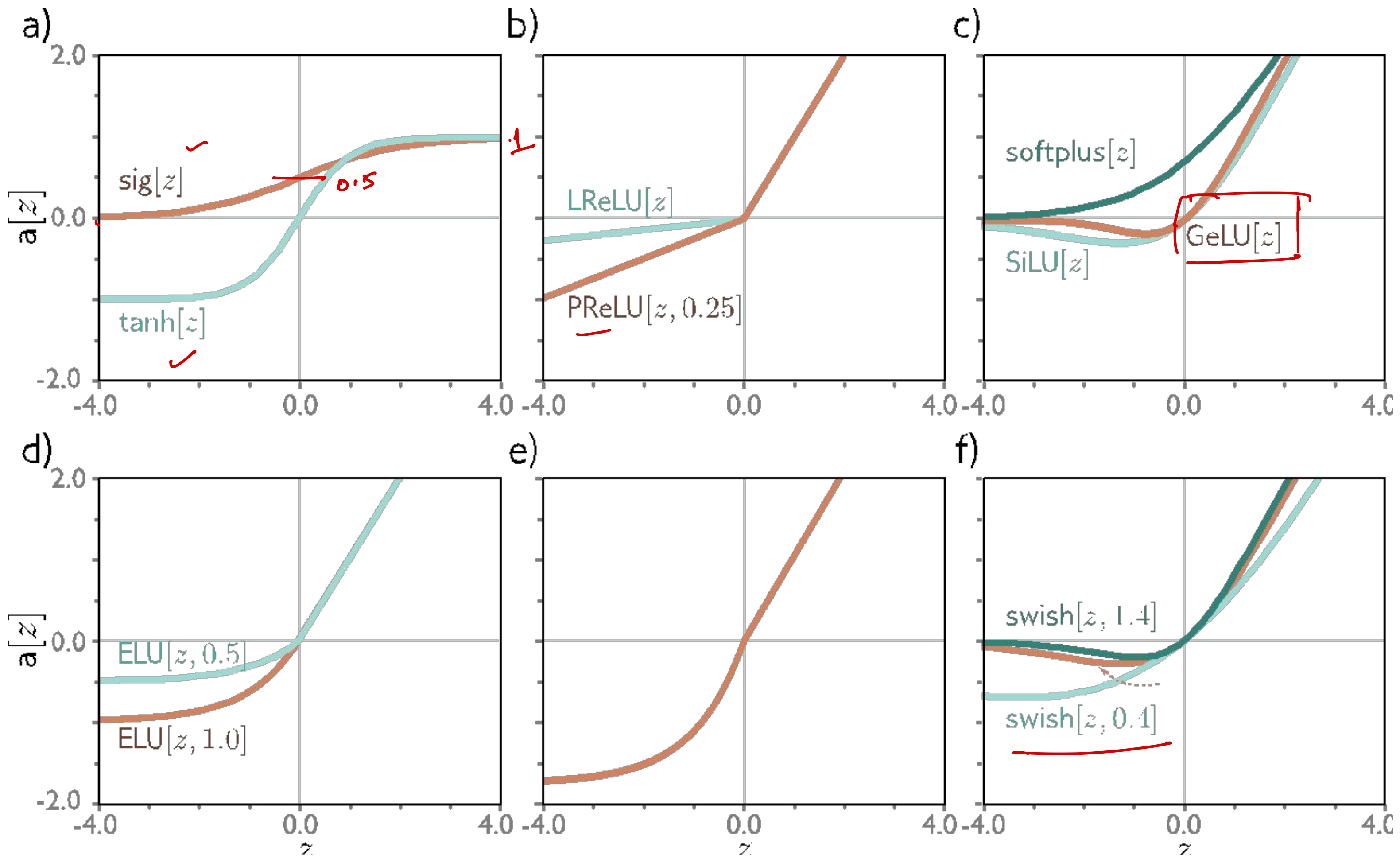
Bias terms are not called the neurons

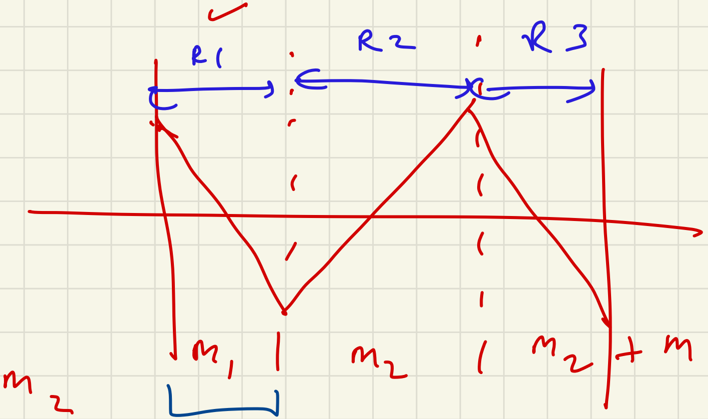
Nomenclature



- Y-offsets = biases
- Slopes = weights
- Everything in one layer connected to everything in the next = fully connected network
- No loops = feedforward network
- Values after ReLU (activation functions) = activations
- Values before ReLU = pre-activations
- One hidden layer = shallow neural network
- More than one hidden layer = deep neural network
- Number of hidden units ≈ capacity

Other activation functions





	R_1	R_2	R_3
n_1	1	0/1	1
n_2		1	

64

m_1 m_2

$m_1 + m_2$

$$a + b\,\phi_1(x) + c\,(\phi_2(x))$$

$\phi_1 h_1$ $\phi_2 h_2$

n_1 n_2

$\alpha_1 x_1 + \beta_1$

$\alpha_2 x_1 + \beta_2$

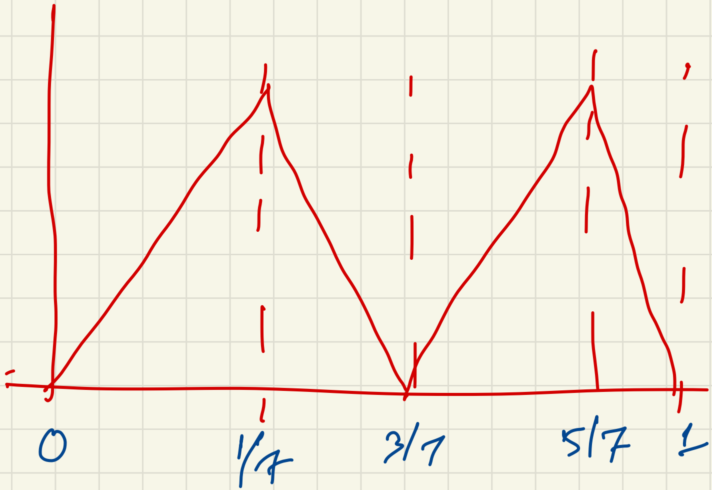
-ve

+ve

$$a + m_1 x + m_2 x$$

0 : Not possible

1 : atmost one joint



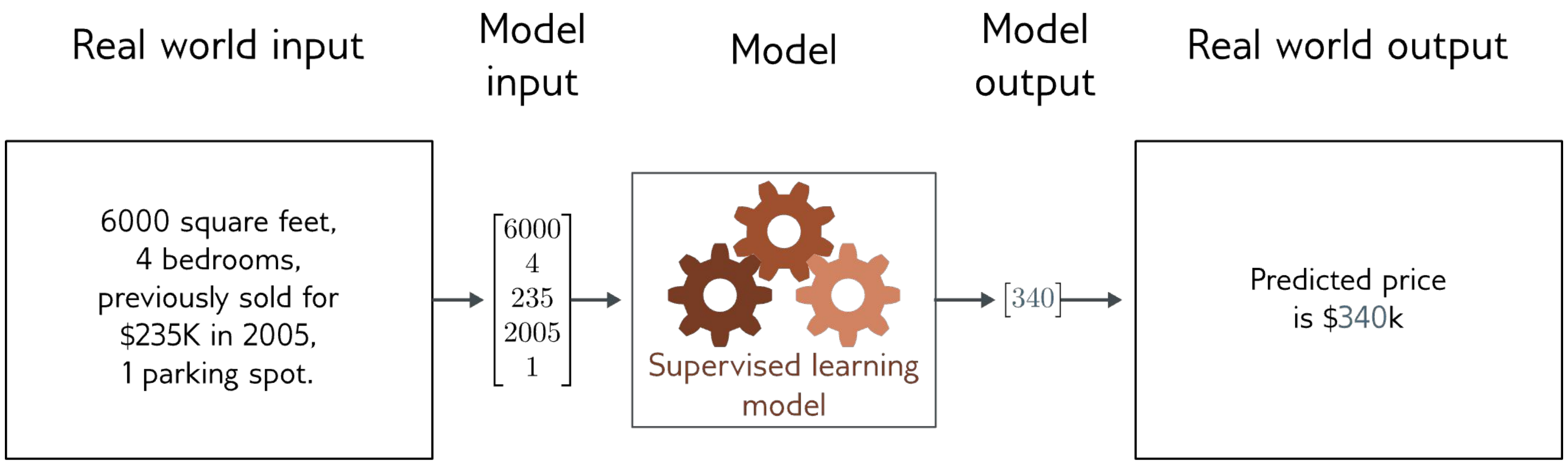
[one possibility]

Can you model this using $\underline{3}$ neurons?

2 -ve
1 +ve

1 -ve
2 +ve

Regression



We have built a model that can:

- take an arbitrary number of inputs
- output an arbitrary number of outputs
- model a function of arbitrary complexity between the two

$$h_d = \mathrm{a}\left[\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di} x_i\right] \qquad y_j = \phi_{j0} + \sum_{d=1}^{D} \phi_{jd} h_d$$