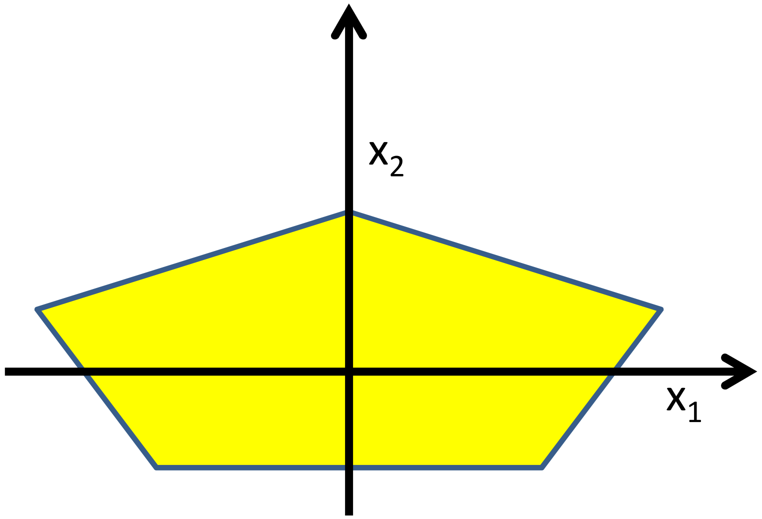
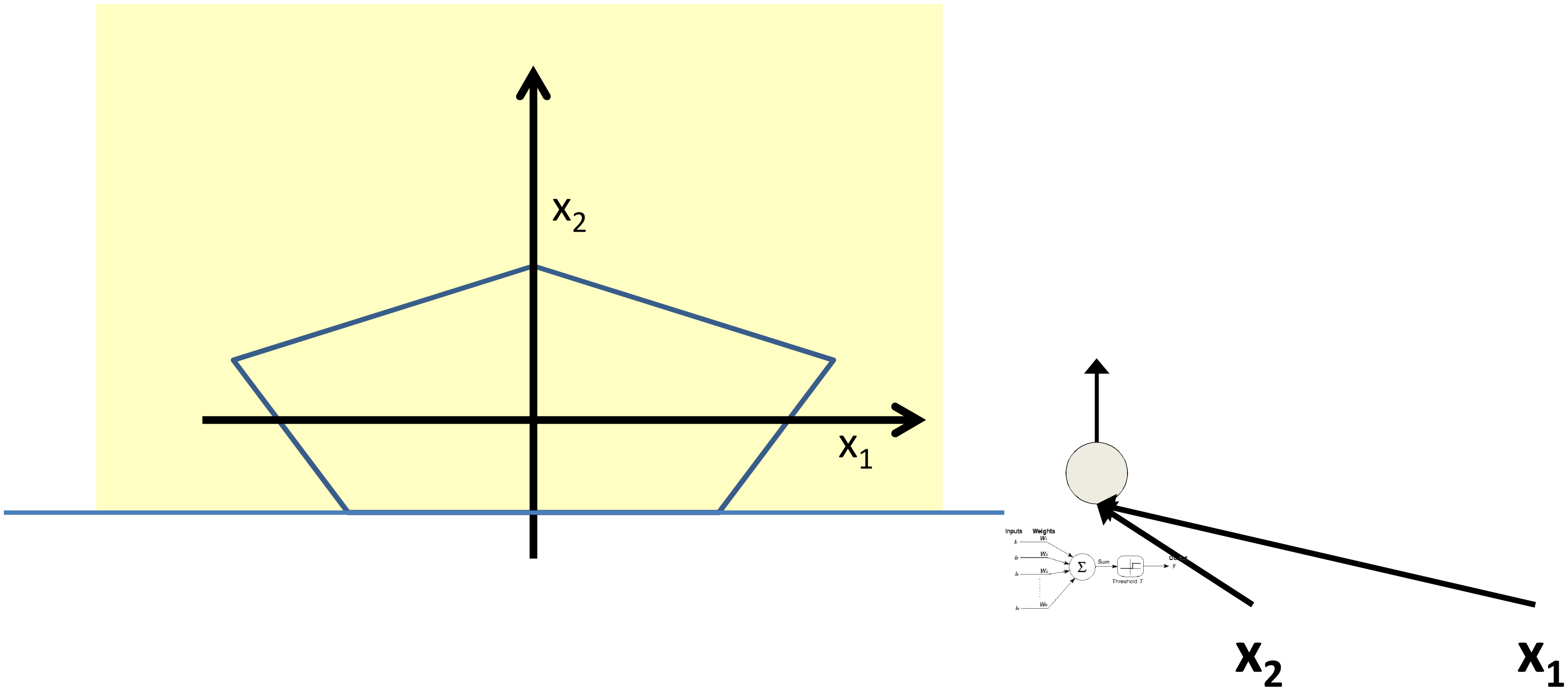
Composing complicated "decision" boundaries



Can now be composed into "networks" to compute arbitrary classification "boundaries"

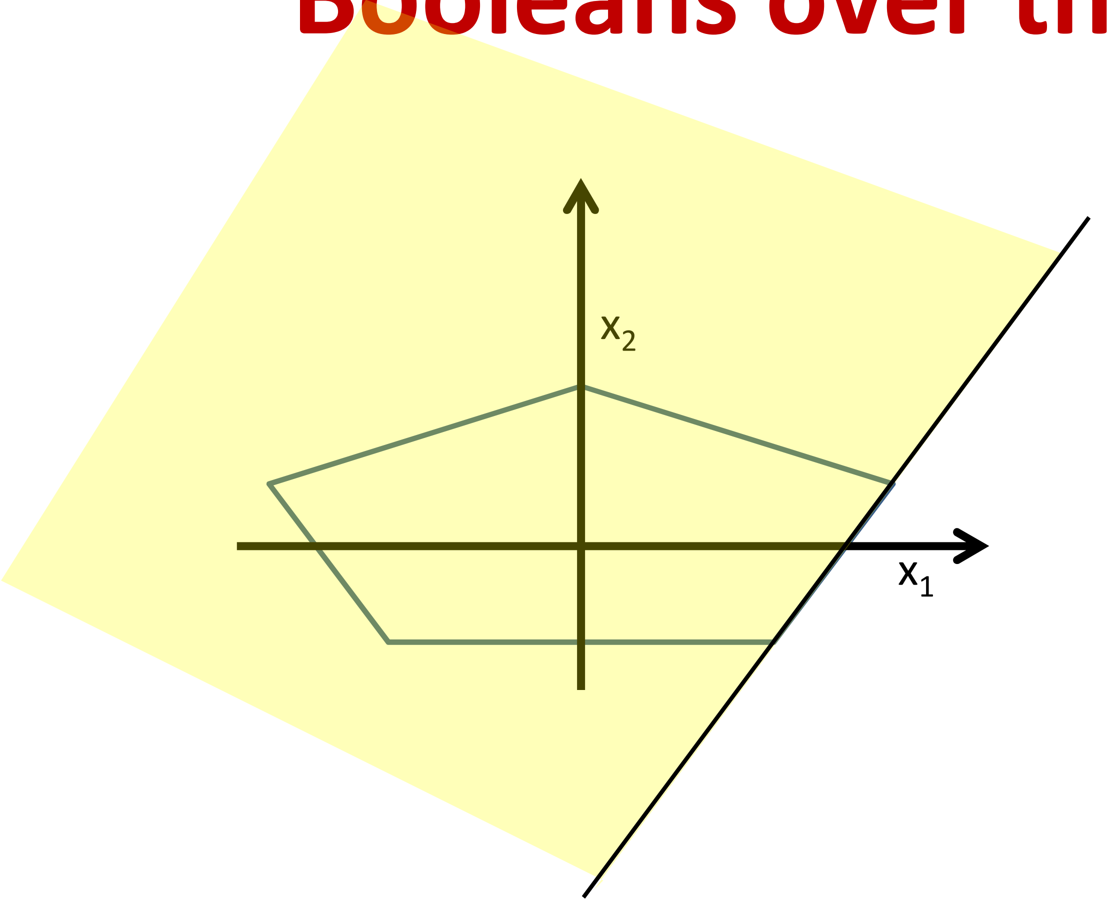
- Build a network of units with a single output that fires if the input is in the coloured area

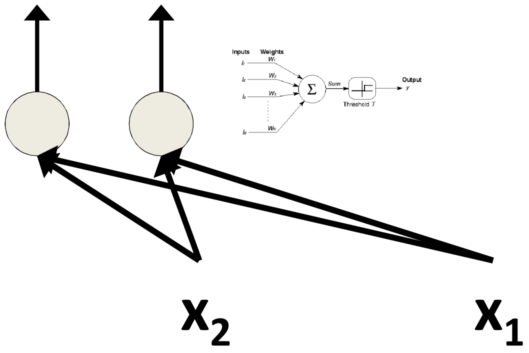
Booleans over the reals



- The network must fire if the input is in the coloured area

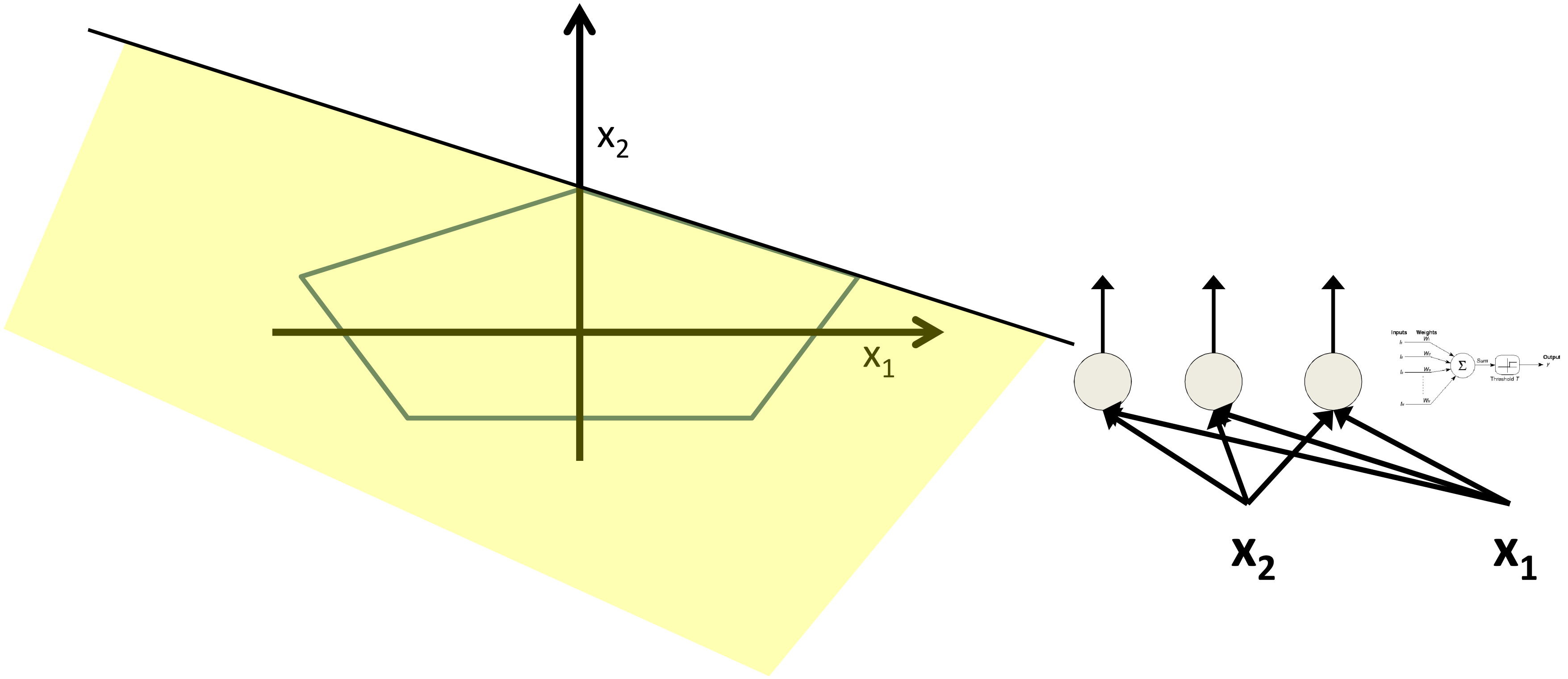
Booleans over the reals





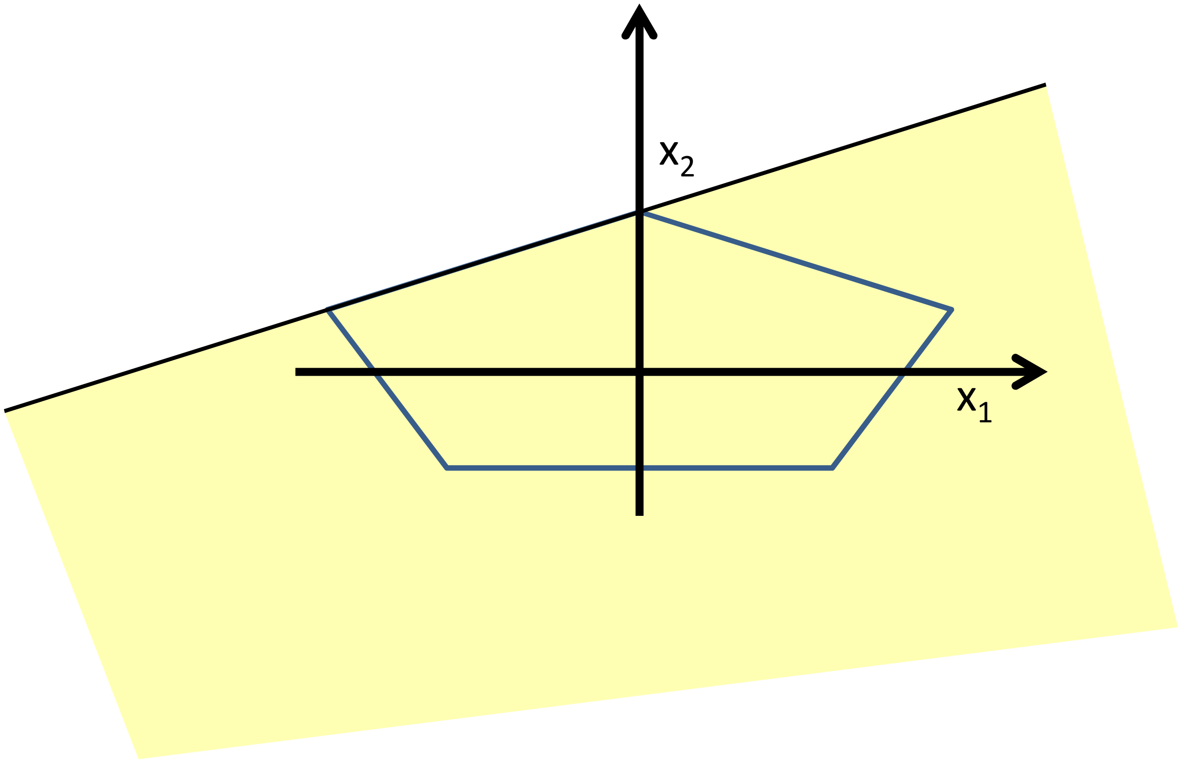
- The network must fire if the input is in the coloured area

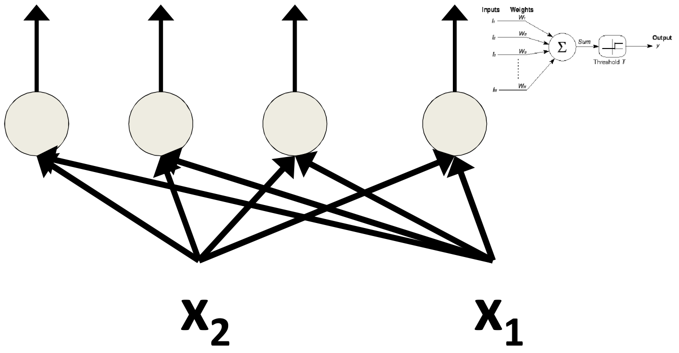
Booleans over the reals



- The network must fire if the input is in the coloured area

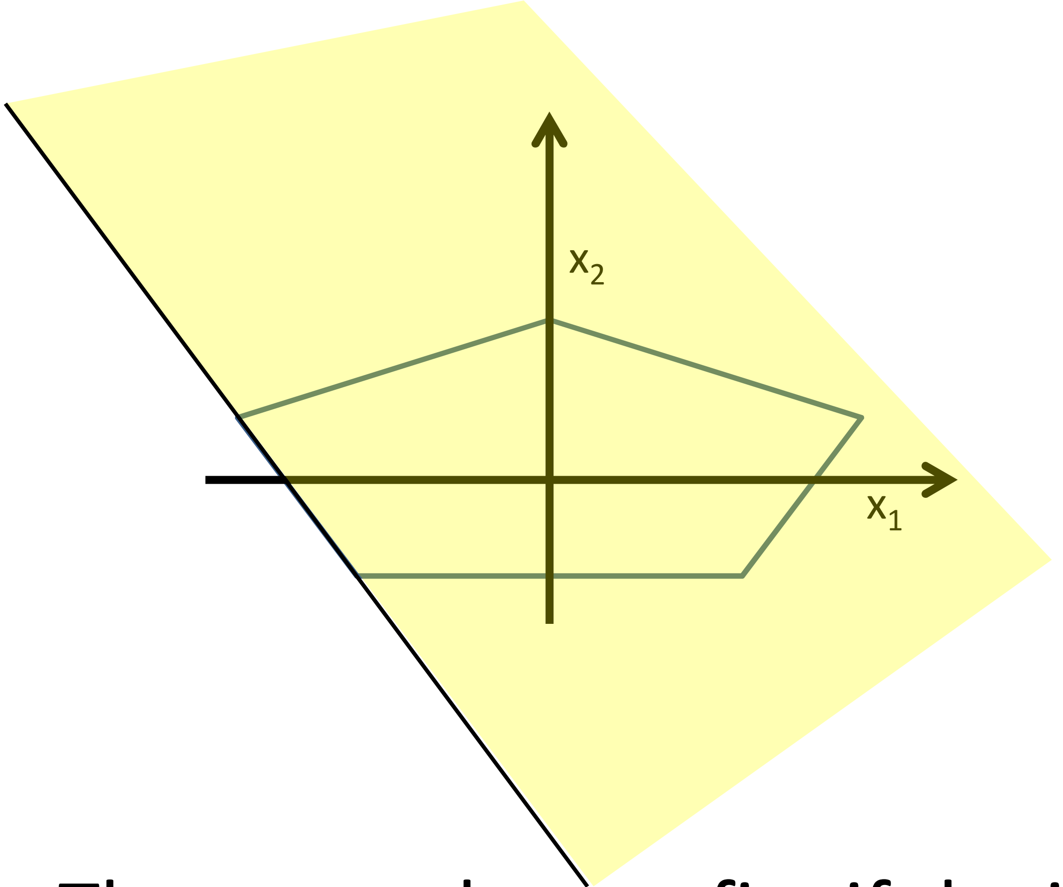
Booleans over the reals

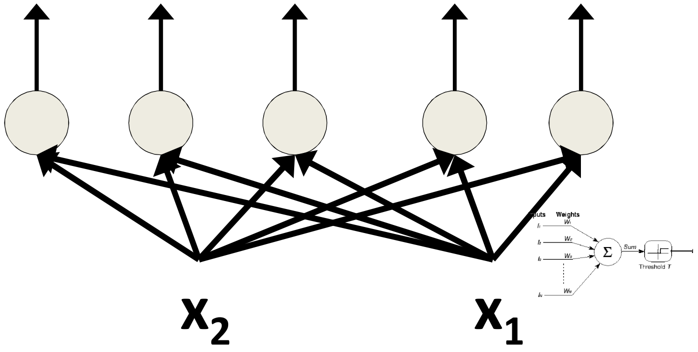




- The network must fire if the input is in the coloured area

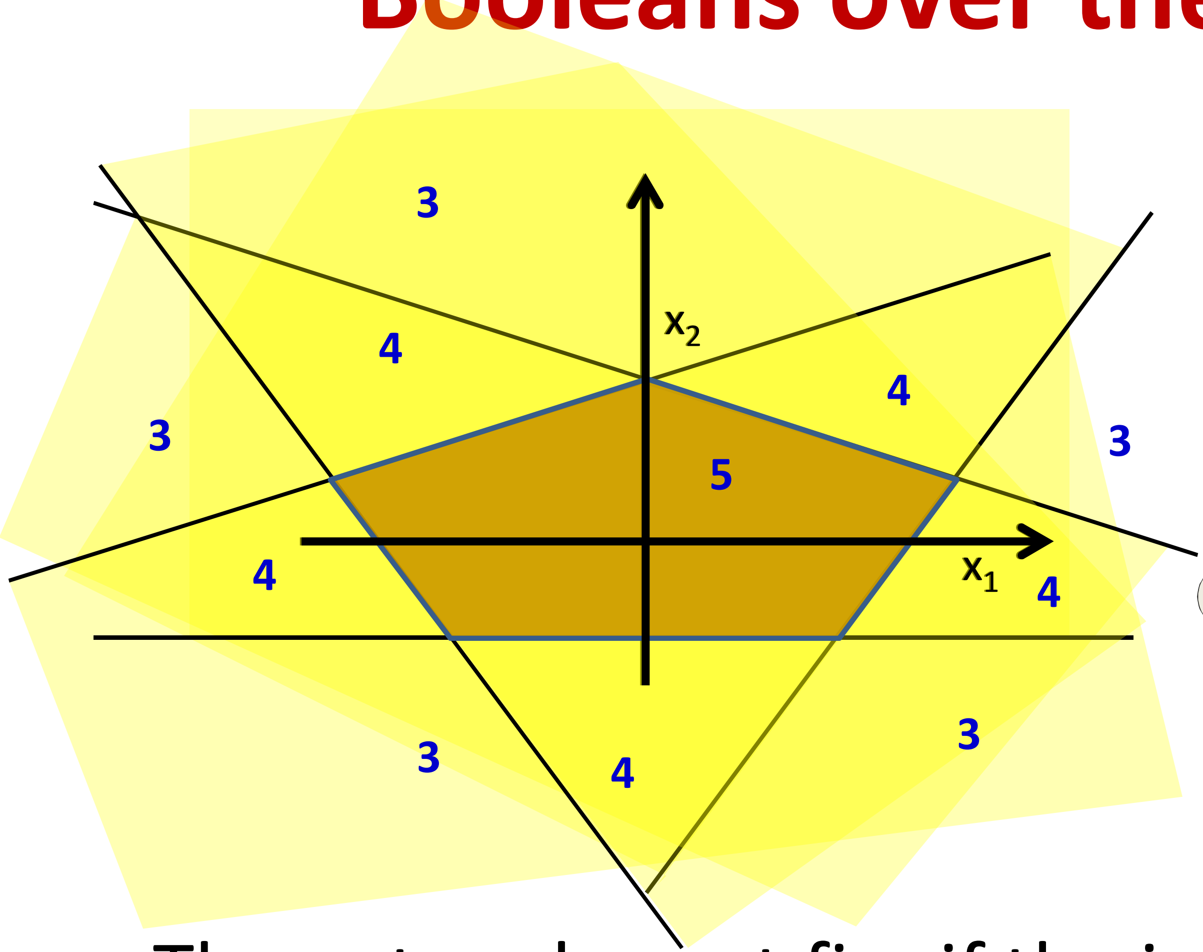
Booleans over the reals

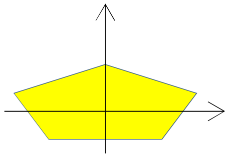




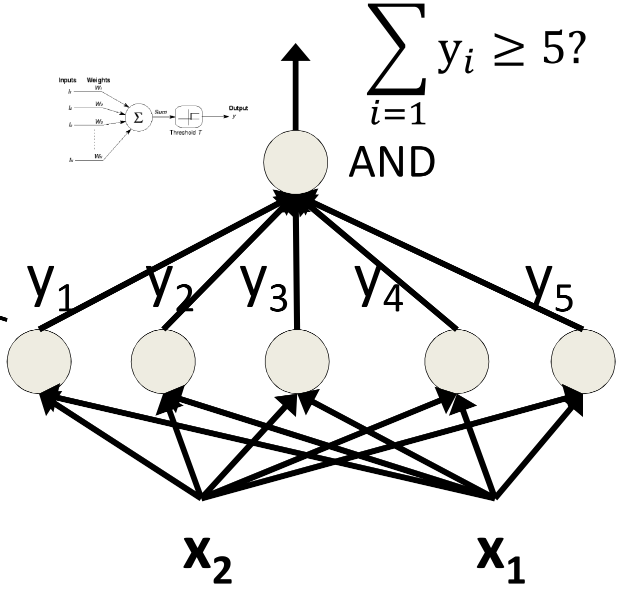
- The network must fire if the input is in the coloured area

Booleans over the reals



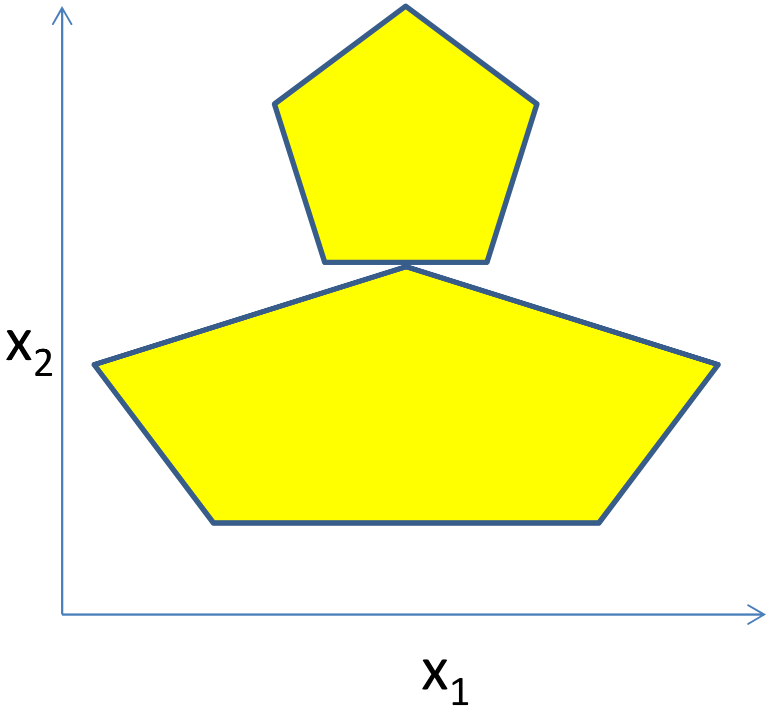


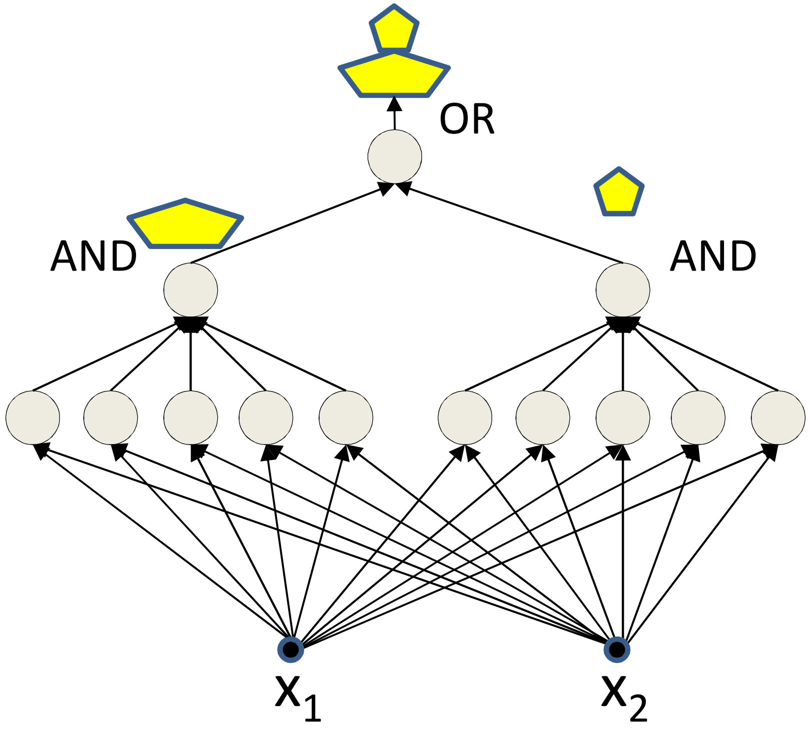
$$\sum_{i=1}^{N} y_i \geq 5?$$



- The network must fire if the input is in the coloured area

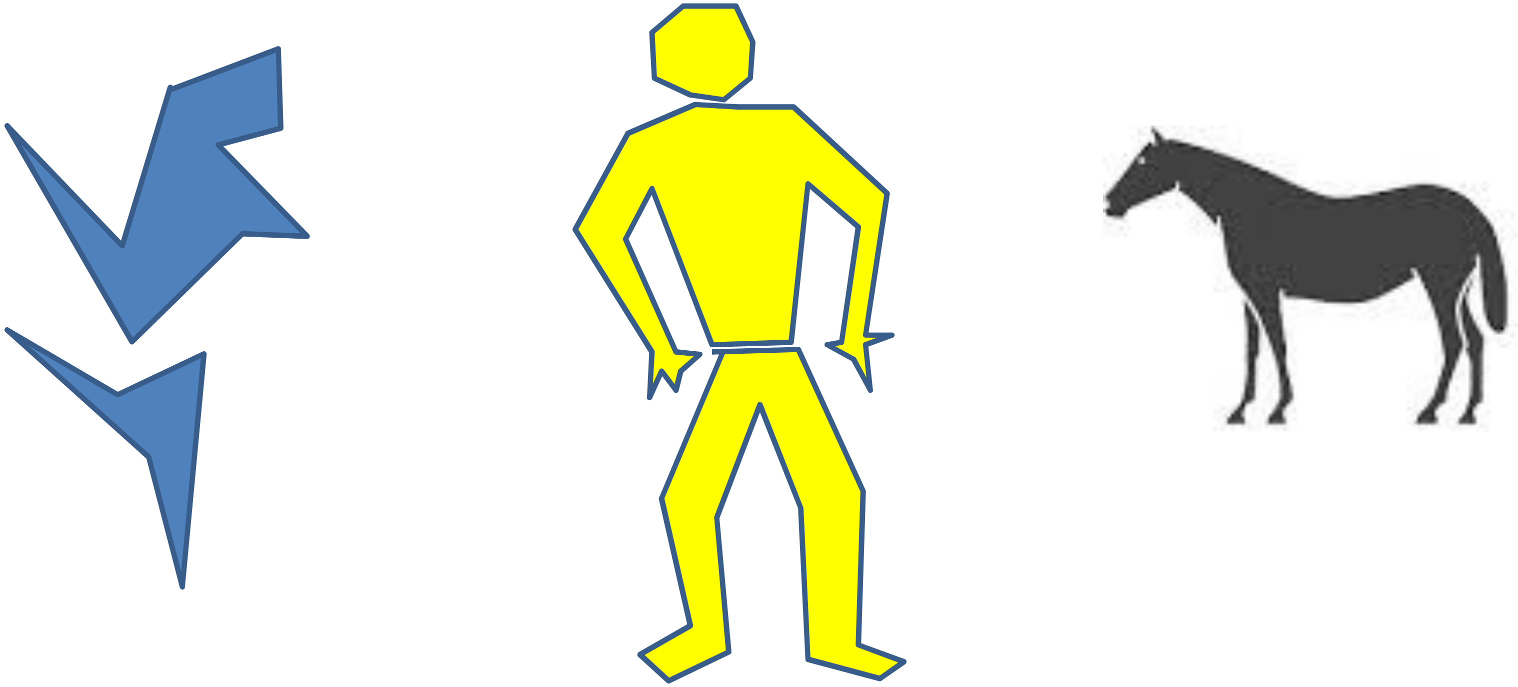
More complex decision boundaries





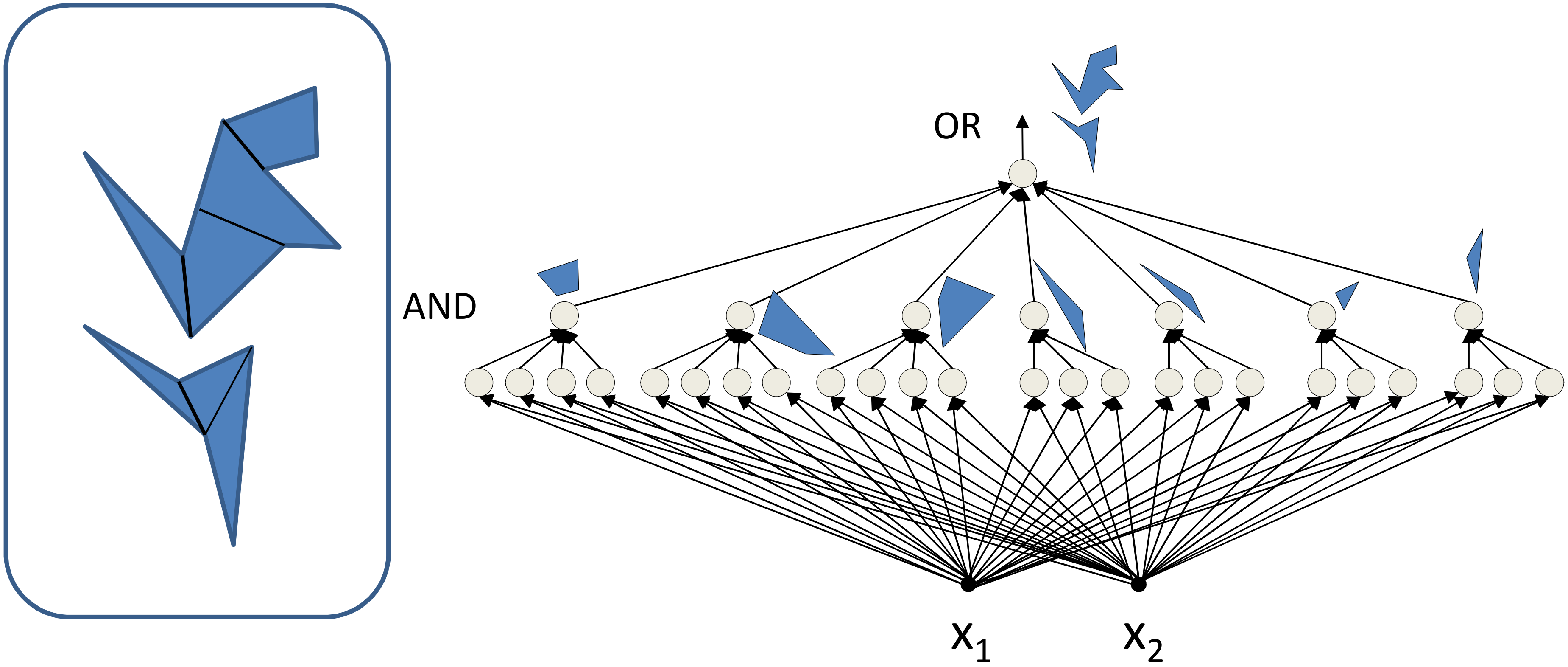
- Network to fire if the input is in the yellow area
  - "OR" two polygons
  - A third layer is required

Complex decision boundaries



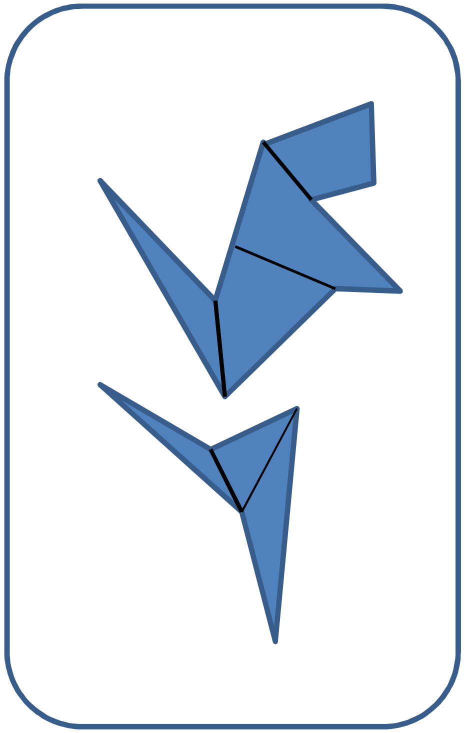
- Can compose *arbitrarily* complex decision boundaries

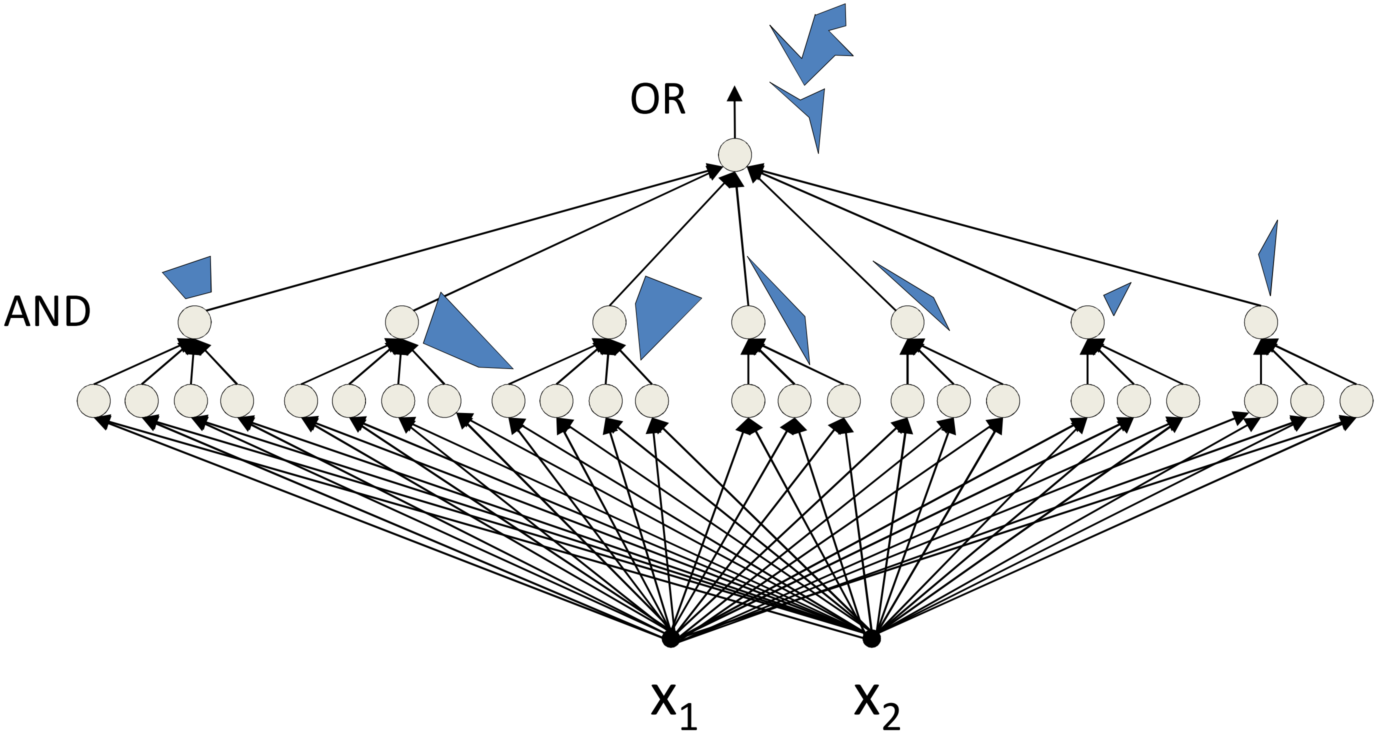
Complex decision boundaries



- Can compose *arbitrarily* complex decision boundaries

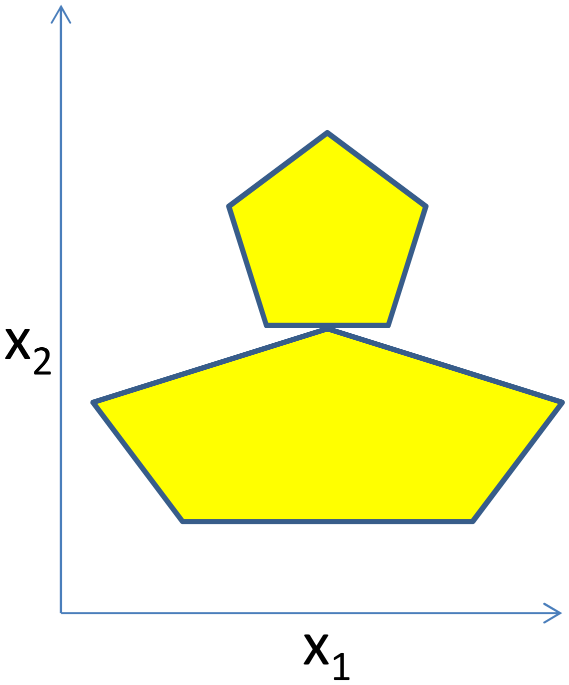
Complex decision boundaries

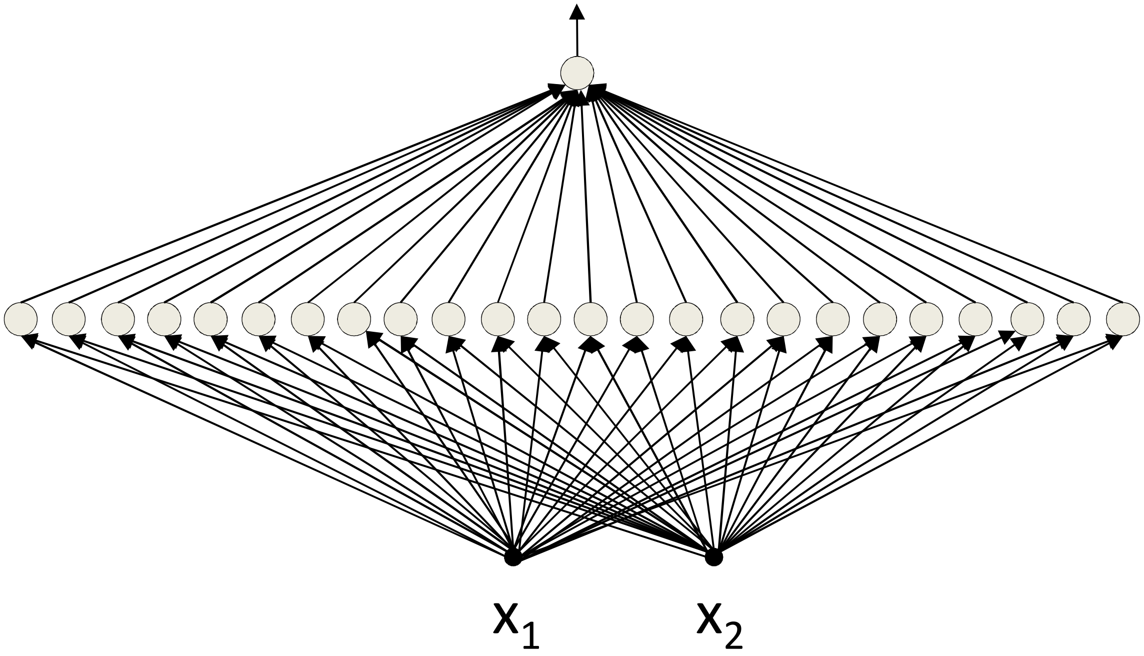




- Can compose *arbitrarily* complex decision boundaries
  - With *only one hidden layer!*
  - **How?**

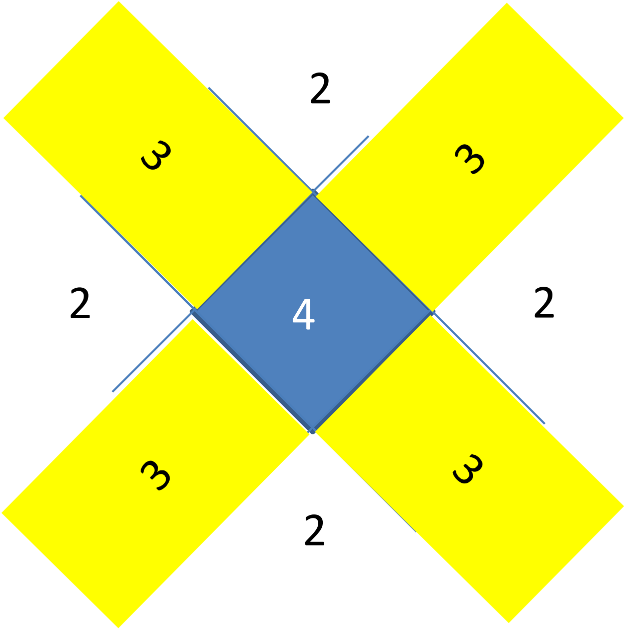
Exercise: compose this with one hidden layer

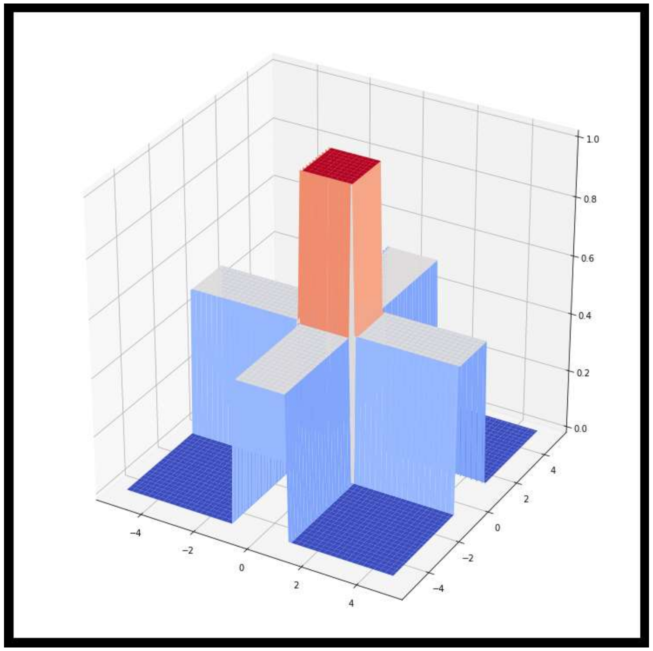




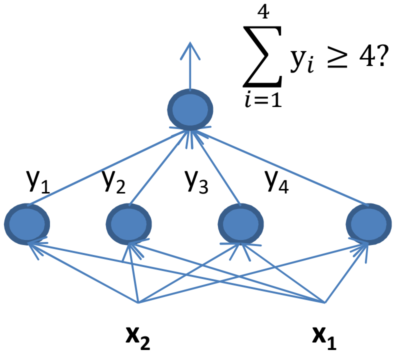
- How would you compose the decision boundary to the left with only *one* hidden layer?

Composing a Square decision boundary

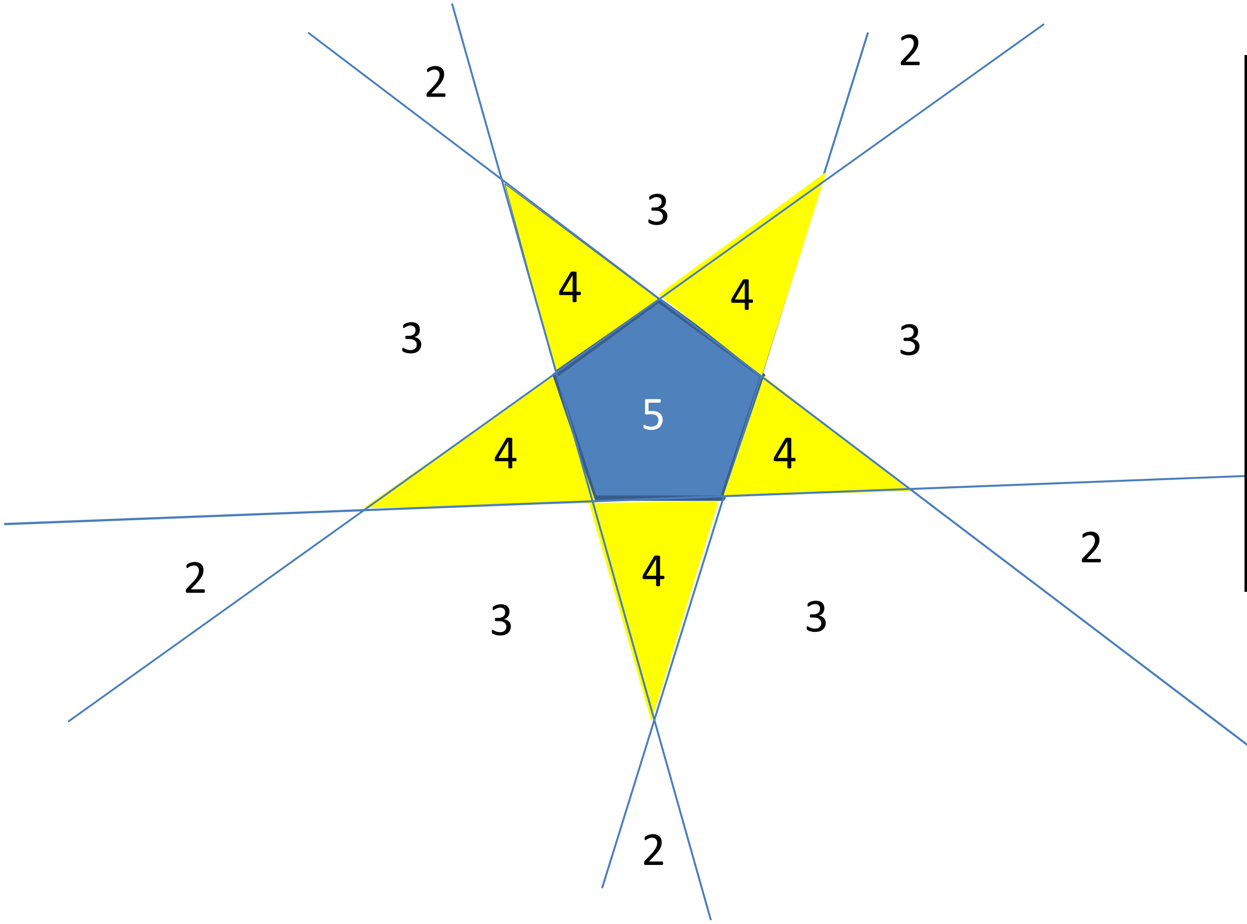


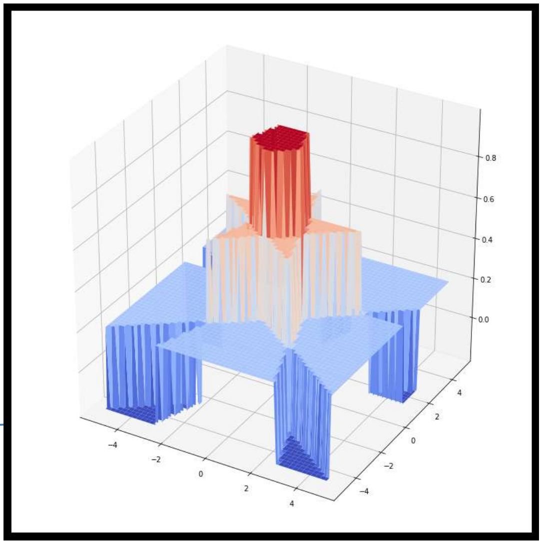


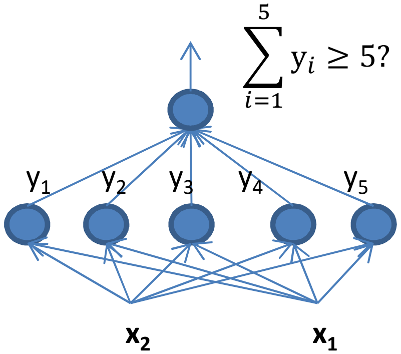
- The polygon net



Composing a pentagon

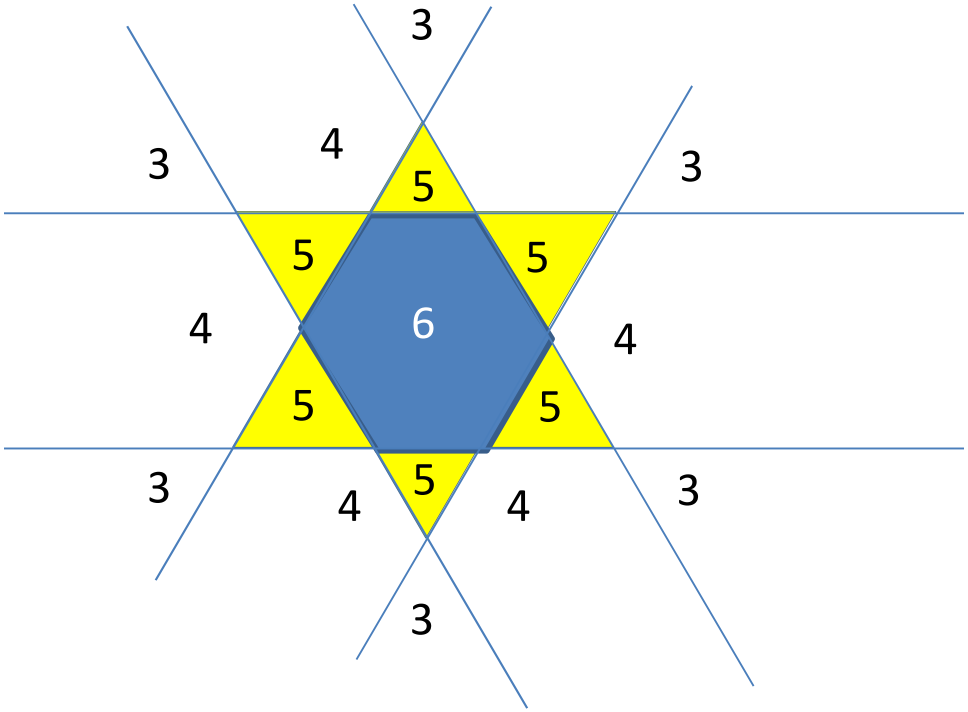


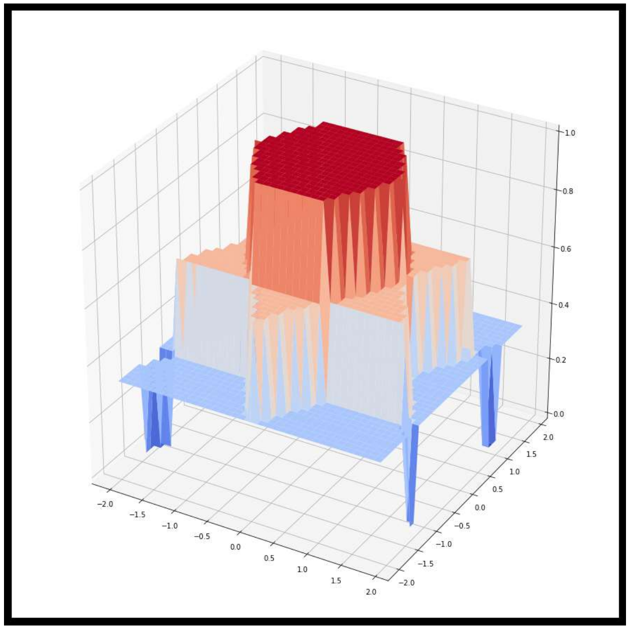




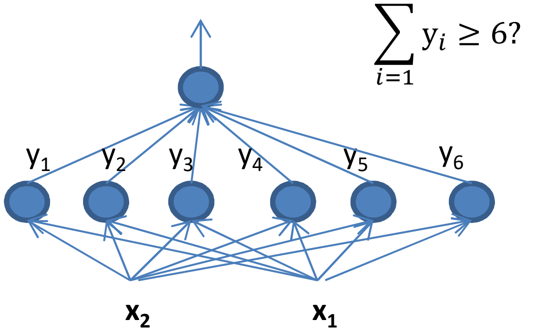
- The polygon net

Composing a hexagon



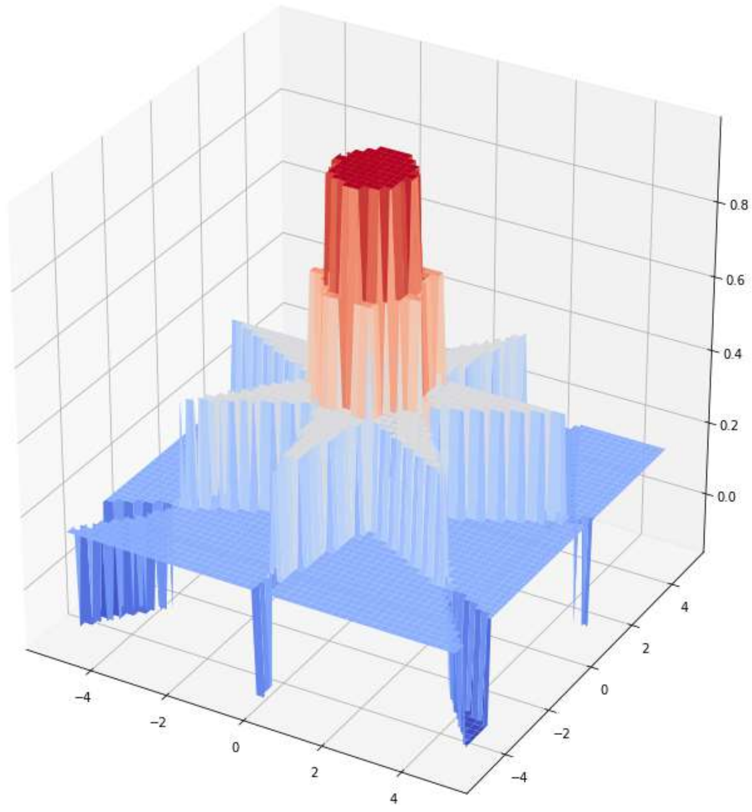


- The polygon net



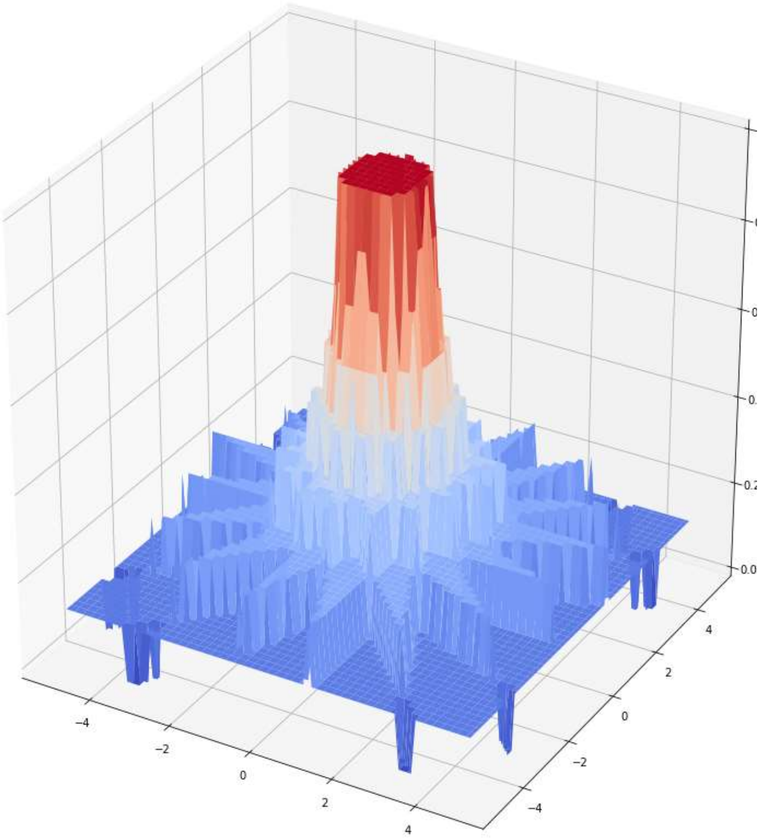
$$\sum_{i=1}^{N} \mathrm{y}_i \geq 6?$$

How about a heptagon



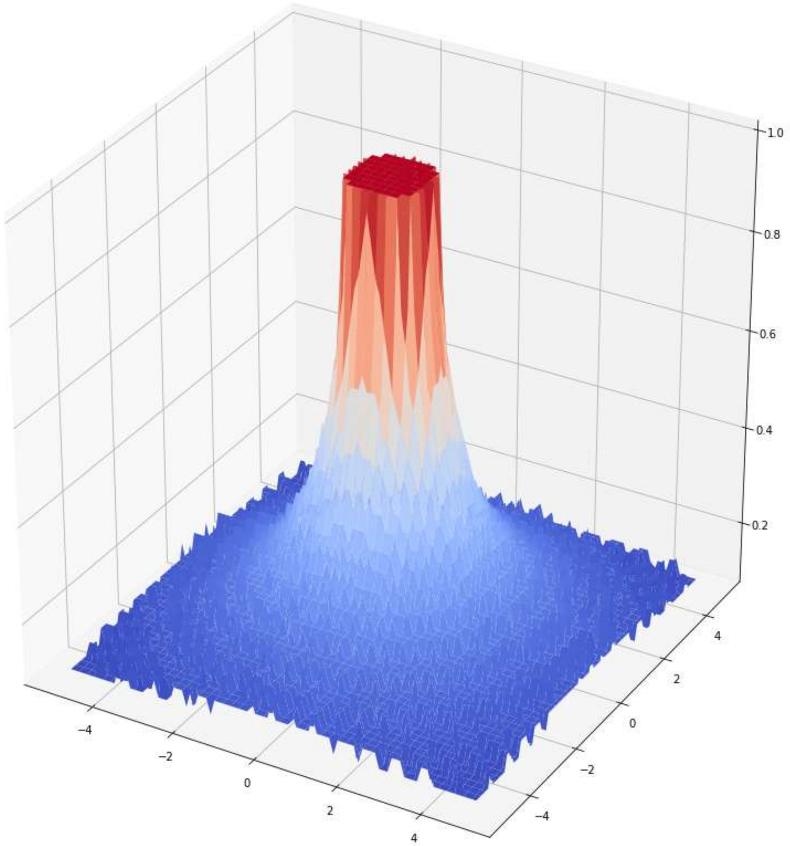
- What are the sums in the different regions?
  - – A pattern emerges as we consider N > 6..
    - N is the number of sides of the polygon

16 sides



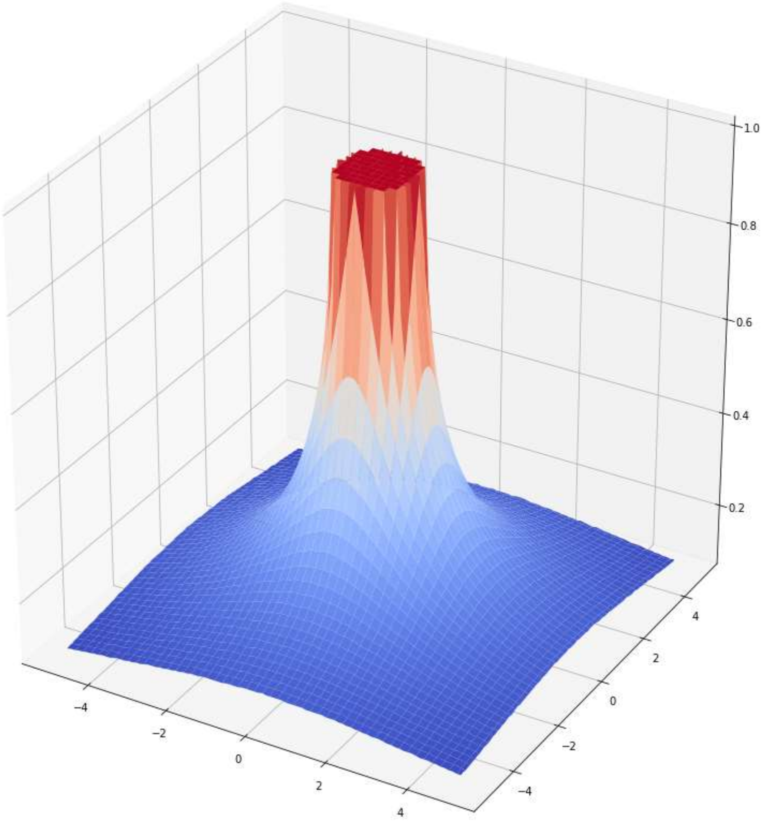
- What are the sums in the different regions?
  - – A pattern emerges as we consider N > 6..

64 sides



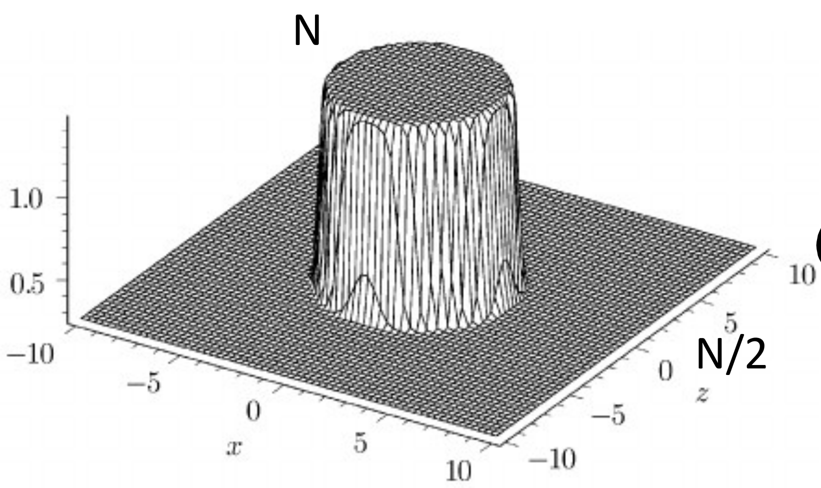
- What are the sums in the different regions?
  - – A pattern emerges as we consider N > 6..

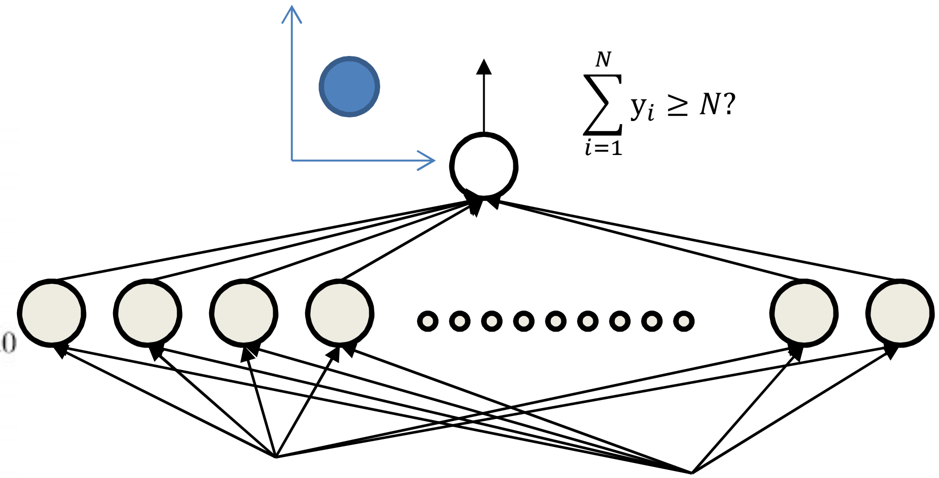
1000 sides



- What are the sums in the different regions?
  - – A pattern emerges as we consider N > 6..

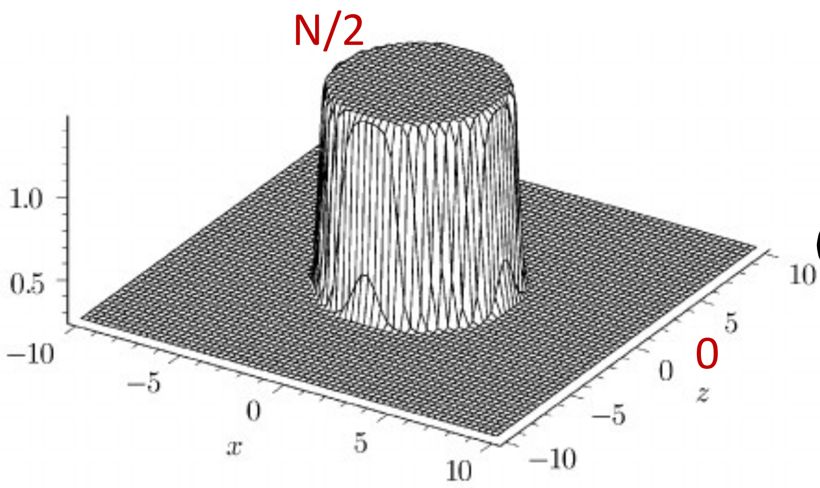
Composing a circle

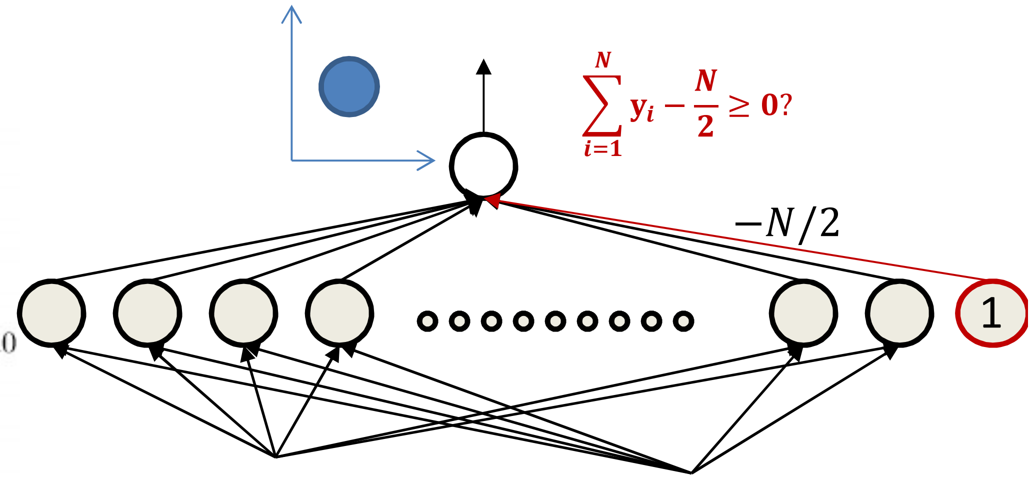




- The circle net
  - Very large number of neurons
  - *Sum is N inside the circle, N/2 outside almost everywhere*
  - Circle can be at any location

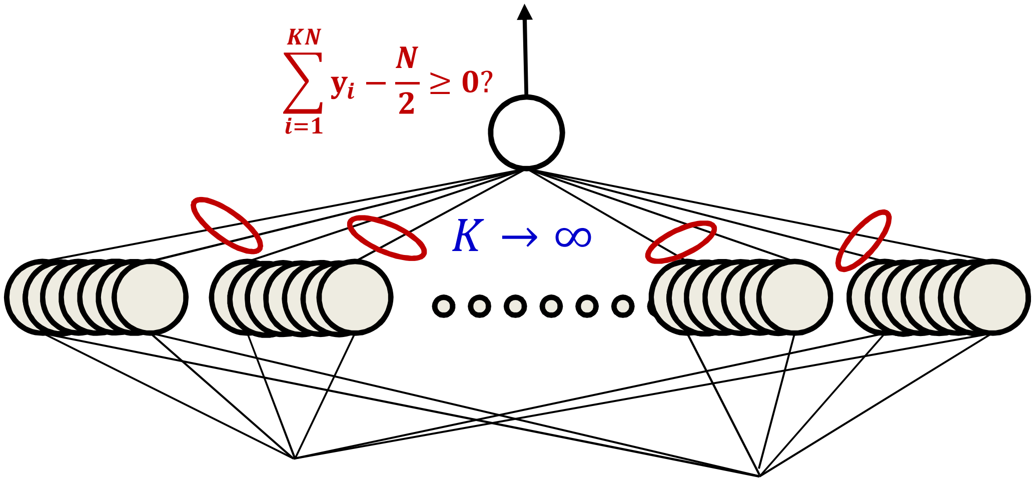
Composing a circle

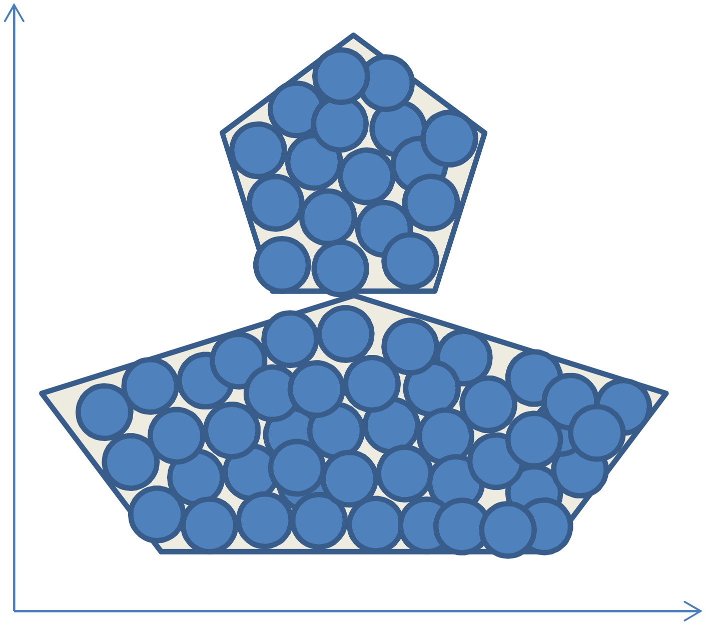




- The circle net
  - Very large number of neurons
  - *Sum is N/2 inside the circle, 0 outside almost everywhere*
  - Circle can be at any location

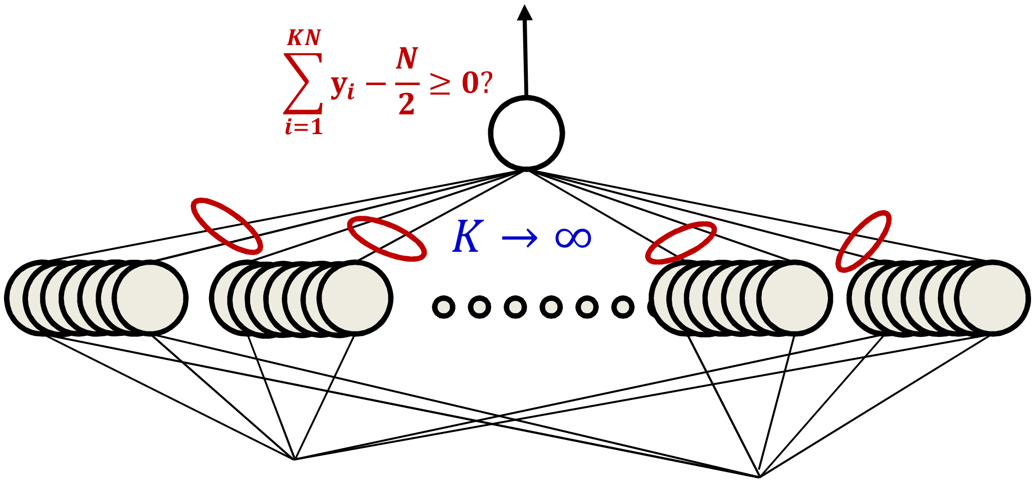
Composing an arbitrary figure

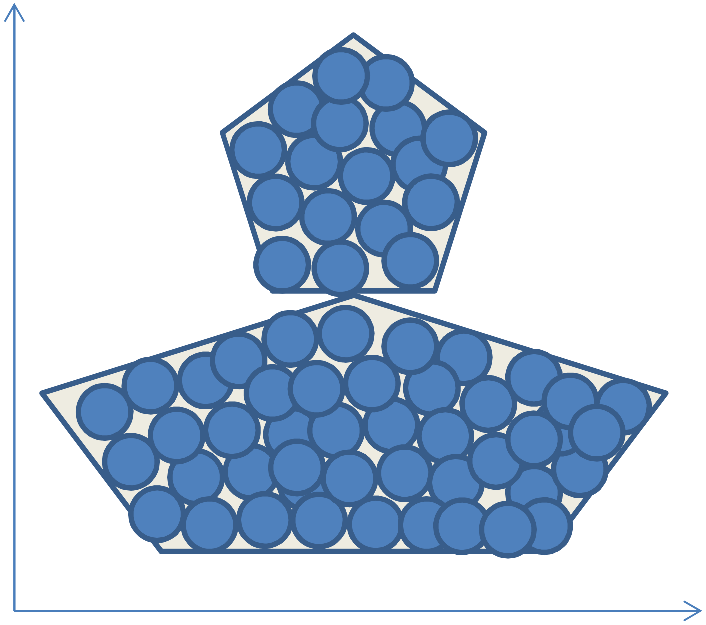




- Just fit in an arbitrary number of circles
  - More accurate approximation with greater number of smaller circles
  - Can achieve arbitrary precision

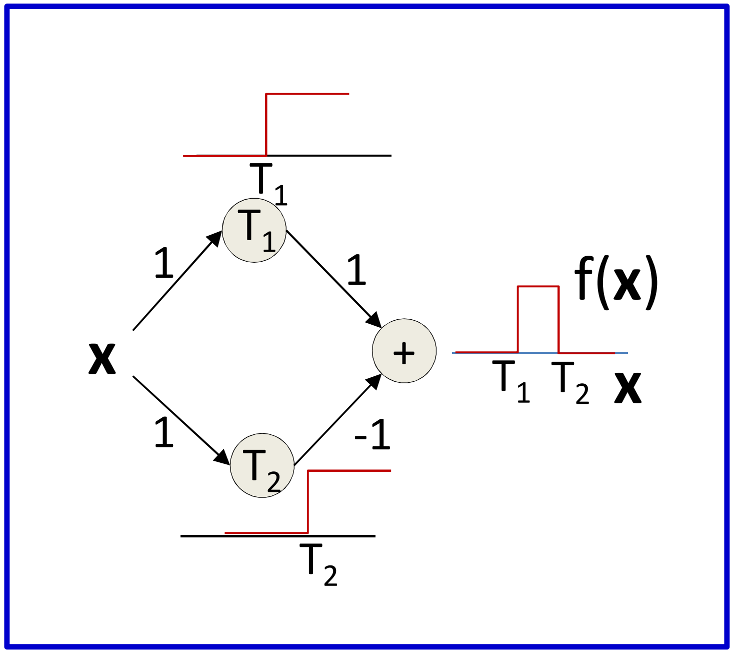
MLP: Universal classifier





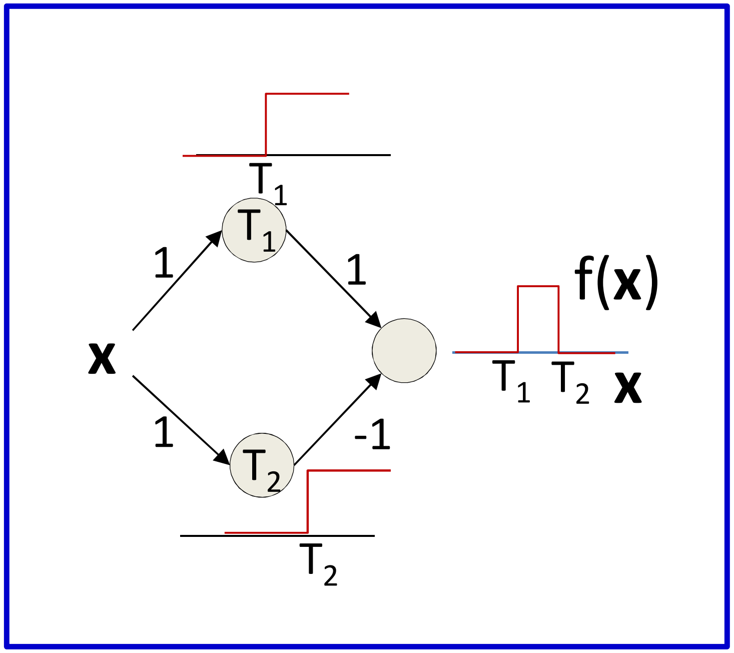
- MLPs can capture *any* classification boundary
- A *one-layer MLP* can model any classification boundary
- *MLPs are universal classifiers*

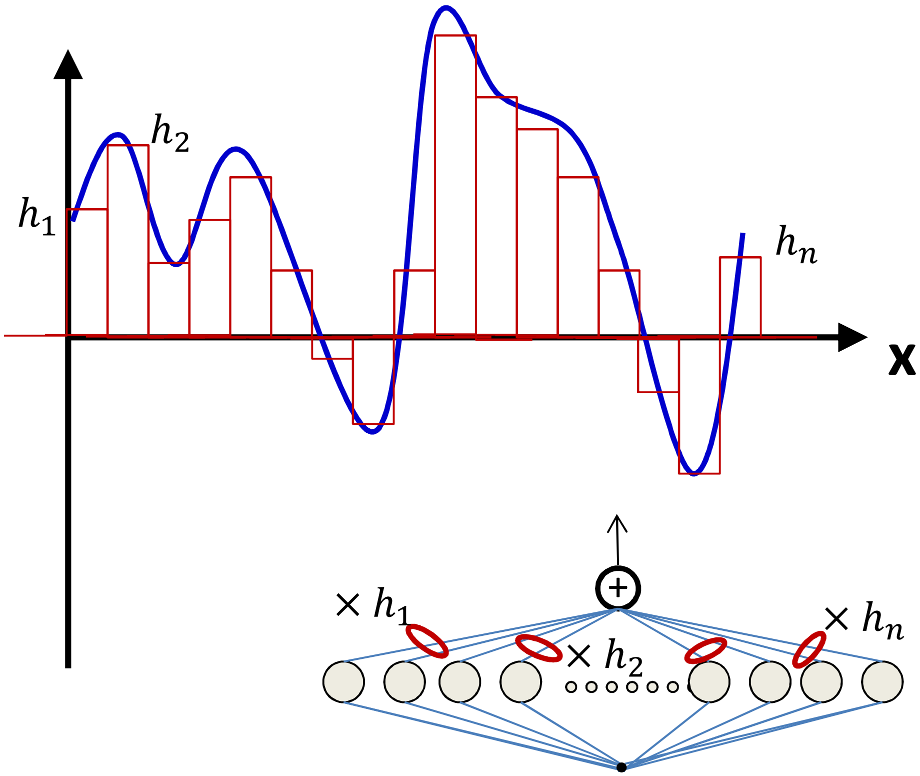
MLP as a continuous-valued regression



- A simple 3-unit MLP with a "summing" output unit can generate a "square pulse" over an input
  - Output is 1 only if the input lies between T_1 and T_2
  - T_1 and T_2 can be arbitrarily specified

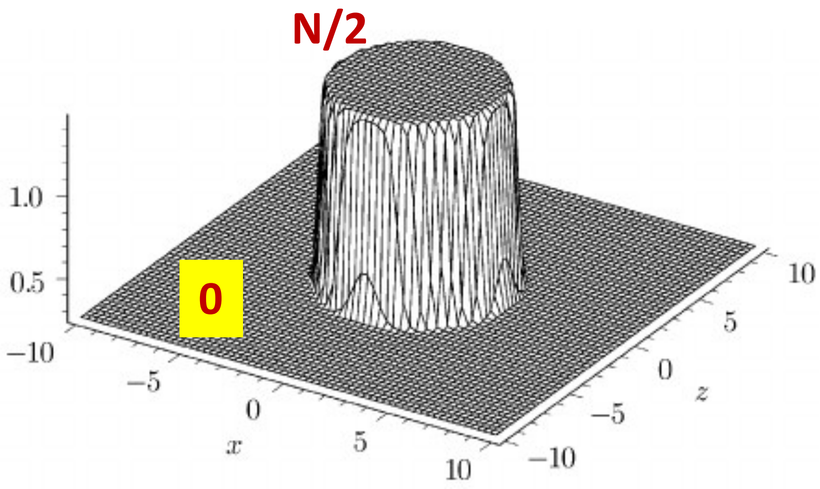
MLP as a continuous-valued regression

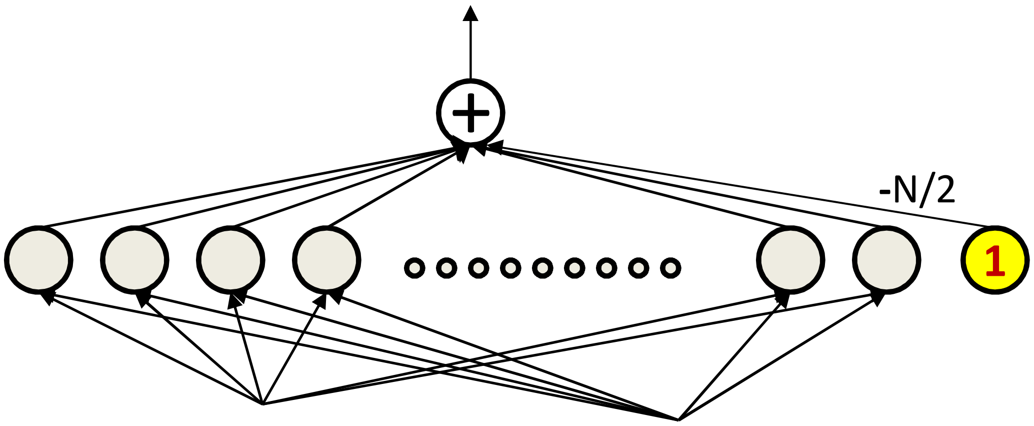




- A simple 3-unit MLP can generate a “square pulse” over an input
- **An MLP with many units can model an arbitrary function over an input**
  - To arbitrary precision
    - Simply make the individual pulses narrower
- ***A one-layer MLP can model an arbitrary function of a single input***

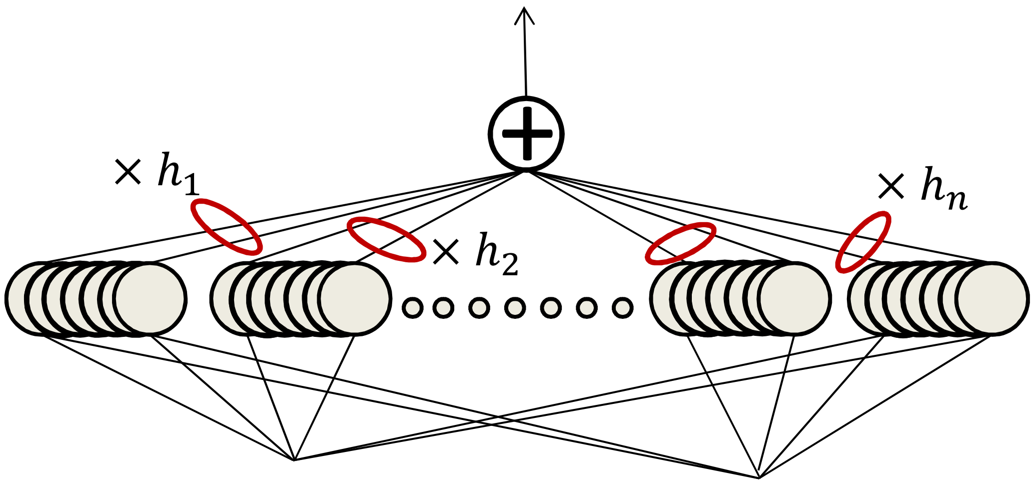
For higher dimensions

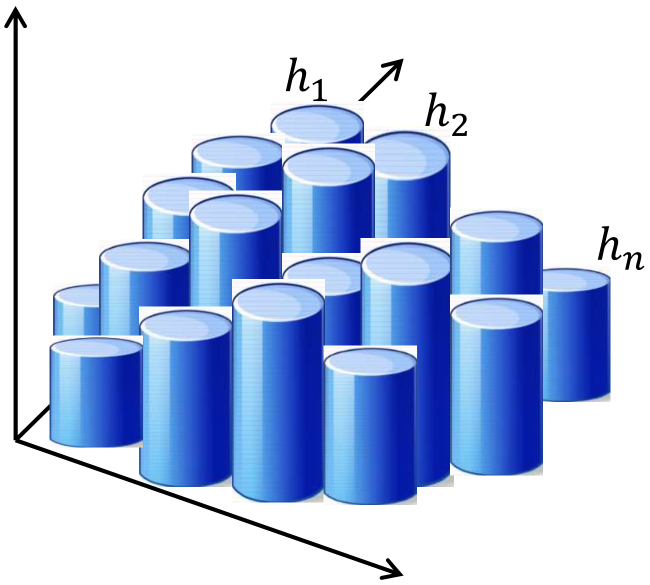




- An MLP can compose a cylinder
  - $N/2$ in the circle, 0 outside

MLP as a continuous-valued function





- MLPs can actually compose arbitrary functions in any number of dimensions!
  - Even with only one layer
    - As sums of scaled and shifted cylinders
  - To arbitrary precision
    - By making the cylinders thinner
  - **The MLP is a universal approximator!**