Deep Learning

Dr. Sugata Ghosal

sugata.ghosal@pilani.bits-pilani.ac.in

BITS Pilani

Pilani Campus
Pilani Campus

innovate achieve lead

BITS Pilani

Pilani Campus

Lecture No. 2| Deep Networks
Time: 11 AM – 1 PM
Date: 05/09/2020

These slides are assembled by the instructor with grateful acknowledgement of the many others who made their course materials freely available online.

How to Specify/Design Perceptron Parameters

x1 OR x2 with Perceptron (Step Threshold)

Perceptron with hard/step threshold
Output = 0 if input w1*x1 + w2*x2 < threshold T
= 1 if input >= T

From OR truth table,
w1*0 + w2*0 < T implies T > 0
w1*1+w2*0 >= T implies w1 >=T
w1*0+w2*1 >= T implies w2 >= T
w1*1+w2*1 >=T implies w1+w2>=T

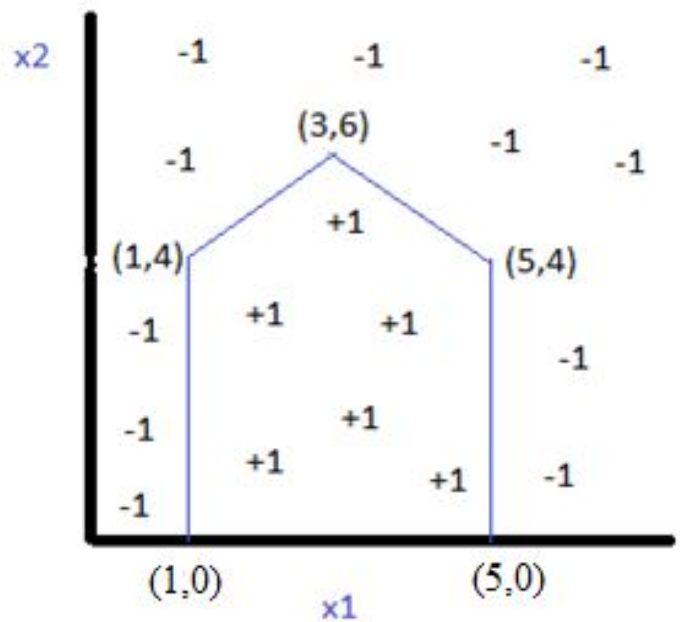
Choose, T =1 (you can choose any T > 0)
Then w1 =1 (you can choose any value >= T)
Similarly choose w2 = 1 (you can choose any value >= T)
w1+w2 >=T is automatically satisfied for w1=w2=1 and T=1

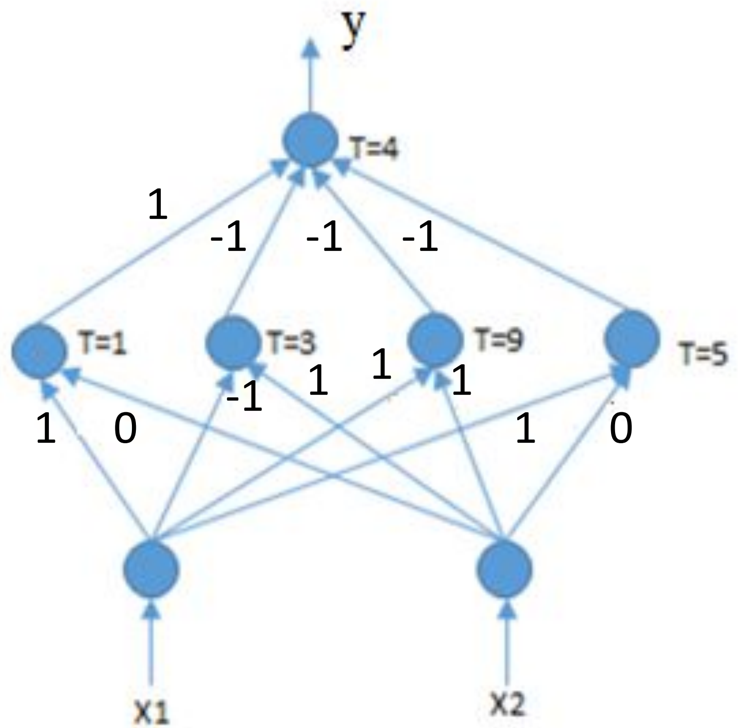
X1 AND x2 Truth Table

x1	x2	Output
0	0	0
1	0	1
0	1	1
1	1	1

Thus, AND can be realized with w1=1, w2=1, T=1 . Important to note that other values of w1, w2, and T can also implement OR, as long as the about 4 inequalities are satisfied

Specify: Perceptron Parameters for Classification



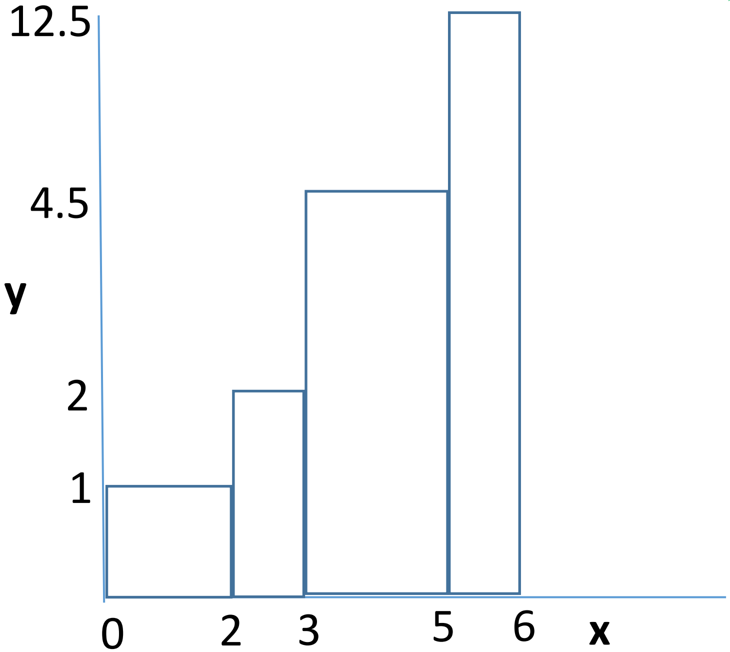


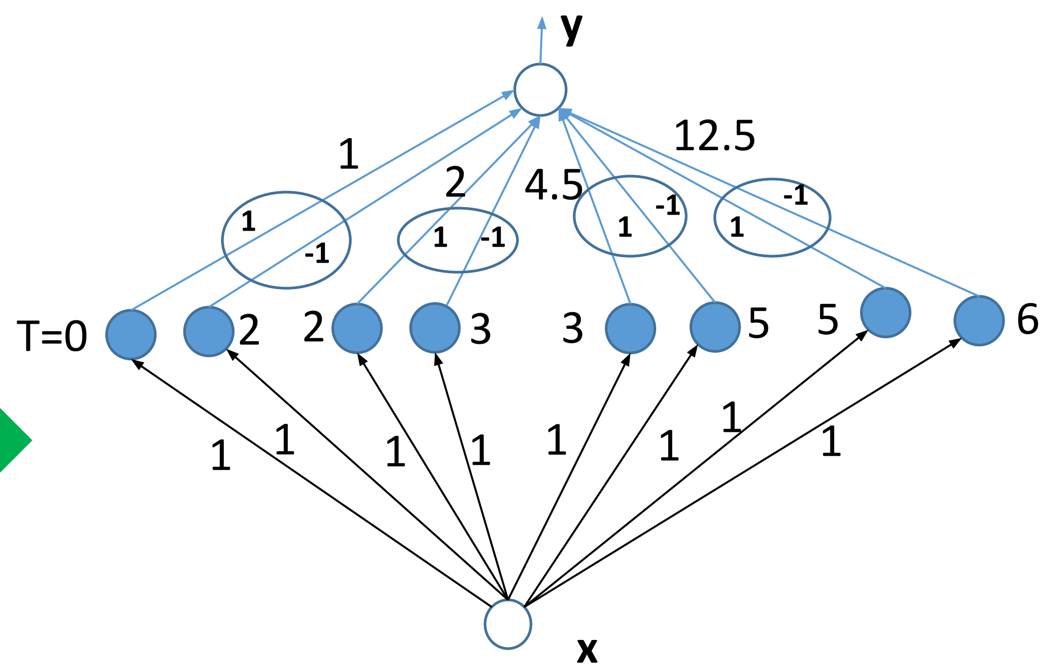
- **Note: Different Choices of weights and bias are possible.**
- Left hidden node implements x1=1 line
- Right hidden node implements x1=5 line
- 2nd node from left implements x2=x1+3 and 3rd from left implements x1+x2=9
- For + class, output of left hidden node = +1, for other nodes output = -1

Example: Functional Approximation in 1-D

Given the function values (x_i, y_i) design an MLP

x	y
0	1
2	2
3	4.5
5	12.5

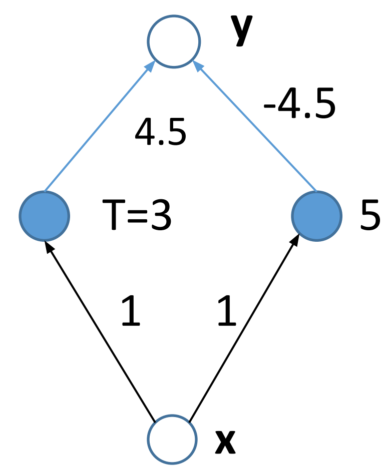




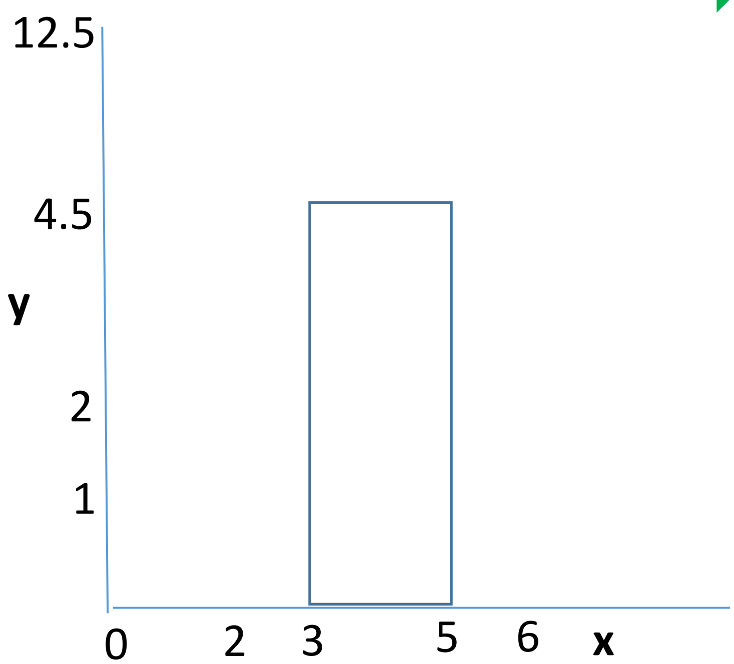
Example: Functional Approximation in 1-D

Given the function values (x_i,y_i) design an MLP

x	y
0	1
2	2
3	4.5
5	12.5



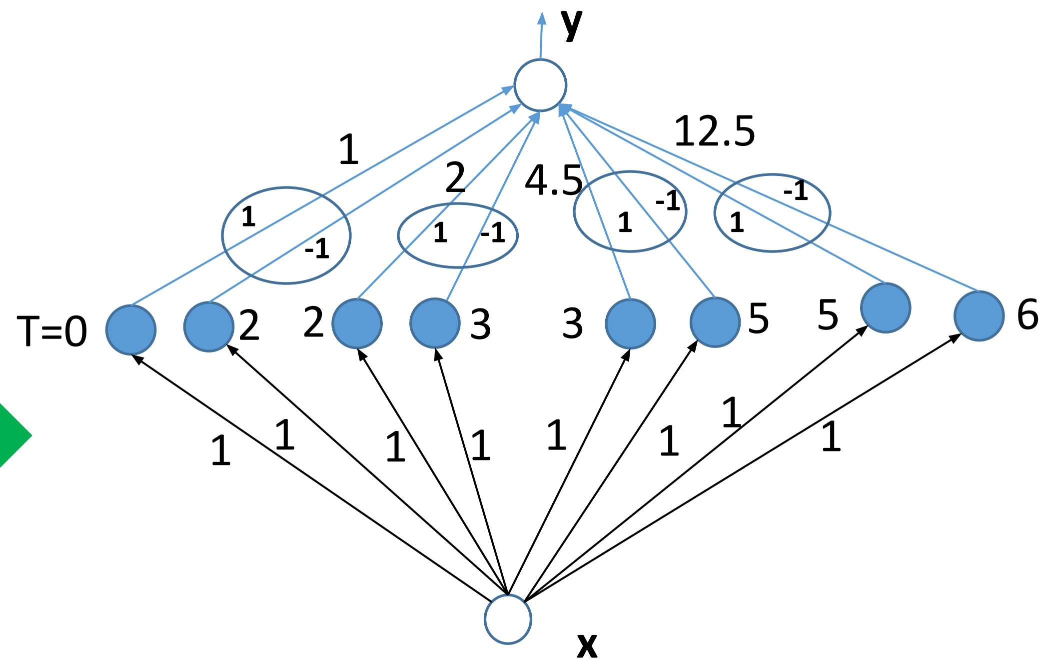
- First consider one data point, say (3, 4.5) and hidden nodes with hard threshold T=3 and T=5 (output = 1 if input >=T else 0)
- Output node uses a ReLU activation, i.e., y = sum of all weighted outputs from hidden nodes
- With choice of weights in the above figure, y = 4.5 for 3<=x<5, and 0 otherwise.
- Note that choice of T=5 is by design, given the set of discrete data points . Any T>=3 up to next data point for the right hidden node would work.

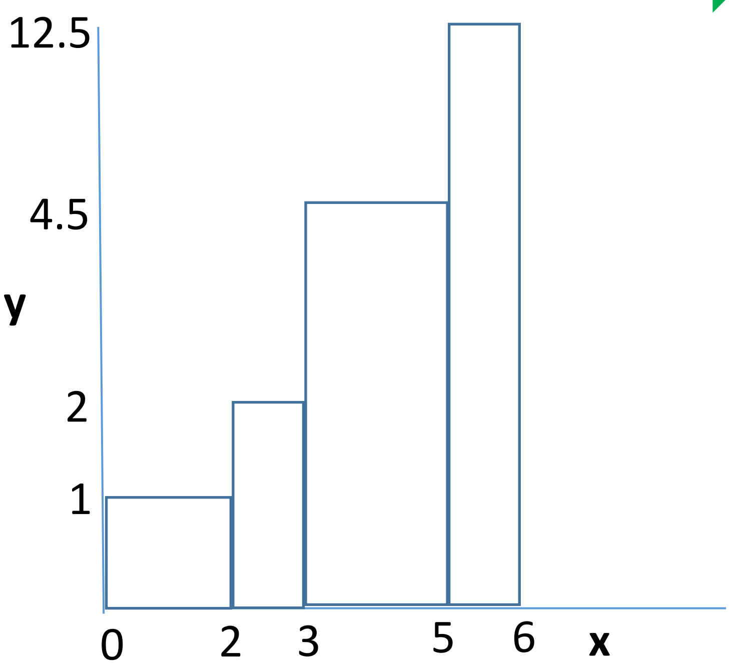


Example: Functional Approximation in 1-D

Given the function values (x_i, y_i) design an MLP

x	y
0	1
2	2
3	4.5
5	12.5





- Applying the method shown in previous slide to all data points, the above network is obtained.
- Output y matches exactly training data points.

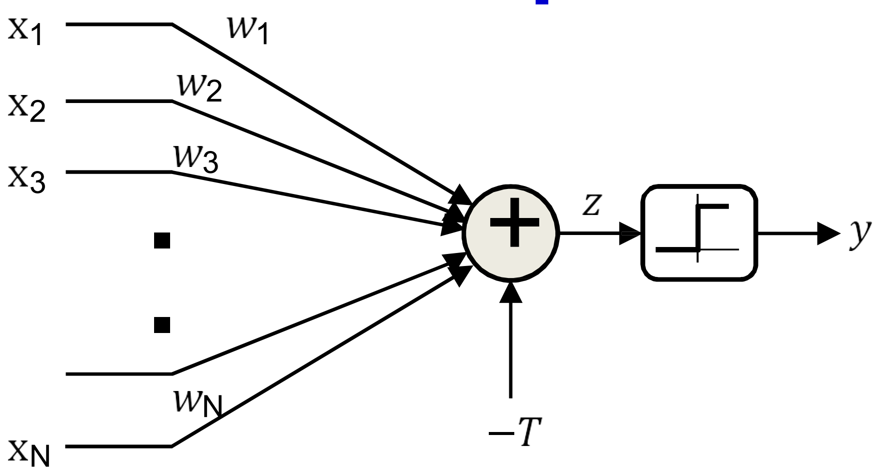
Agenda

- Soft Perceptron
- Multilayer Perceptron as
    - Universal Boolean Function
    - Universal Classifier
    - Universal Function Approximation

- Error Minimization for training

- Optimization Refresher
- Computational Graph

Recap: Perceptron

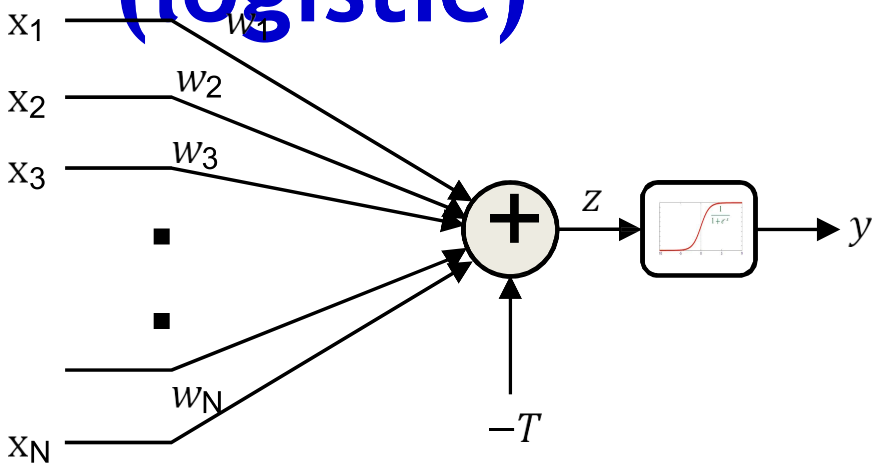


$$z = \sum_i w_i x_i - T$$

$$y = \begin{cases} 1 \; if \; z \geq 0 \\ 0 \; else \end{cases}$$

- A threshold unit
  - “Fires” if the weighted sum of inputs and the “bias” T is positive

The "soft" perceptron (logistic)

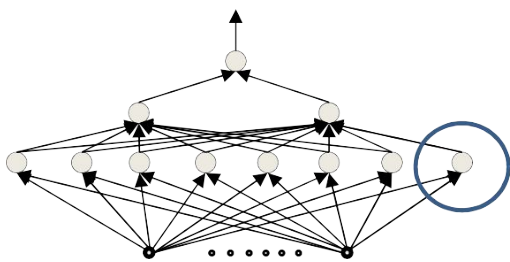


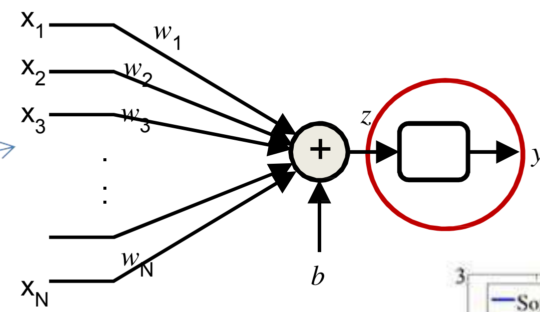
$$z = \sum_i w_i x_i - T$$

$$y = \frac{1}{1 + exp(-z)}$$

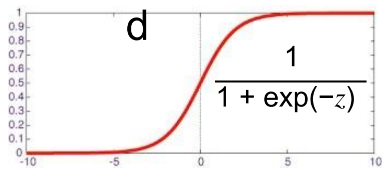
- A "squashing" function instead of a threshold at the output
  - The sigmoid "activation" replaces the threshold
    - **Activation:** The function that acts on the weighted combination of inputs (and threshold)

Other “activations”

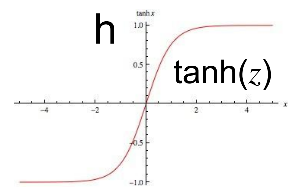


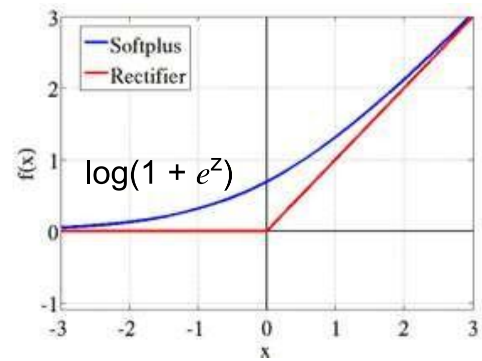


sigmoi
d



tan
h

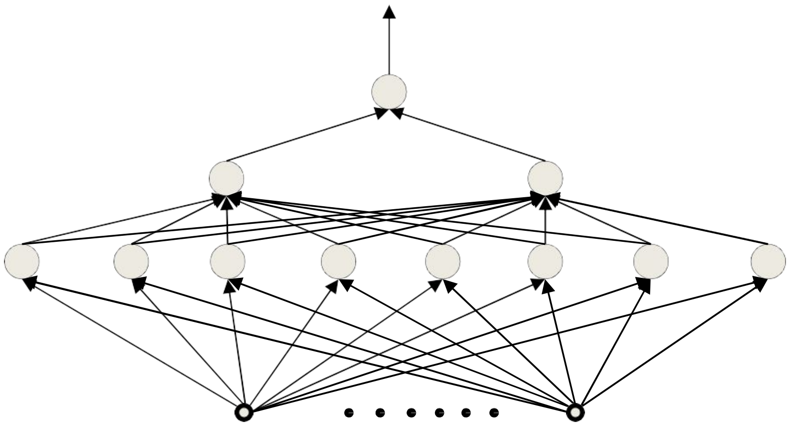


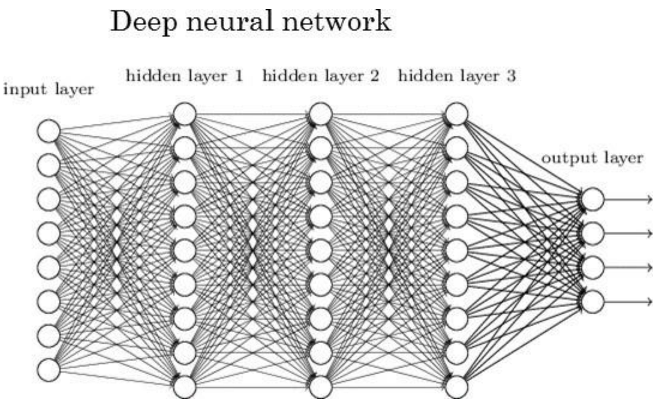


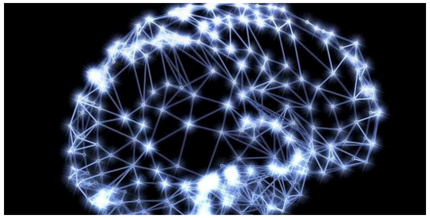
- Does not always have to be a squashing function
  - We will hear more about activations later
- We will continue to assume a “threshold” activation in this lecture

Neural Networks: What can a network represent

The *multi-layer* perceptron

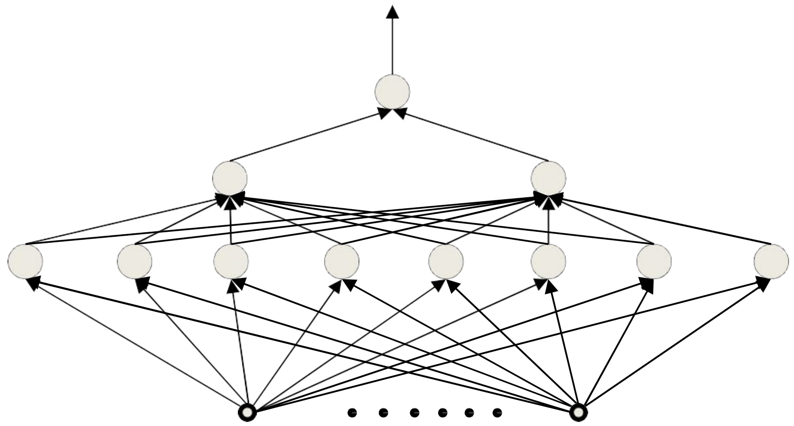


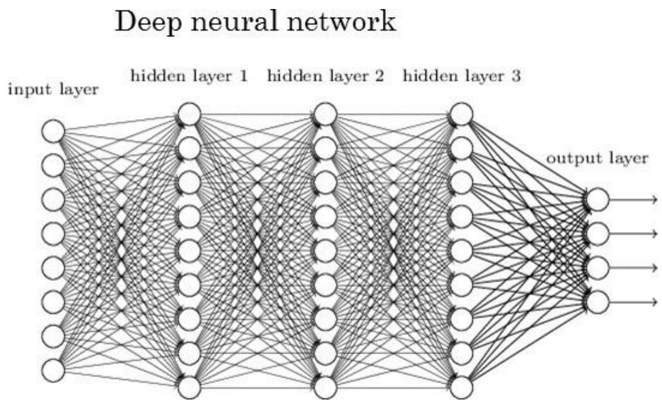




- A network of perceptrons
  - Generally "layered"

Defining “depth”

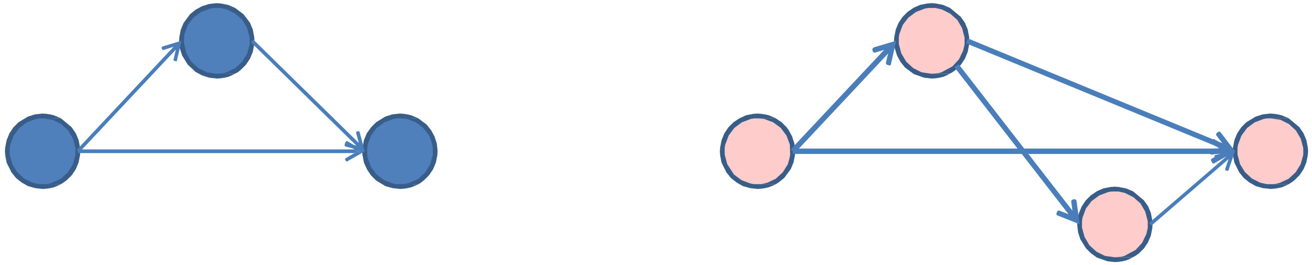




- What is a “deep” network

Deep Structures

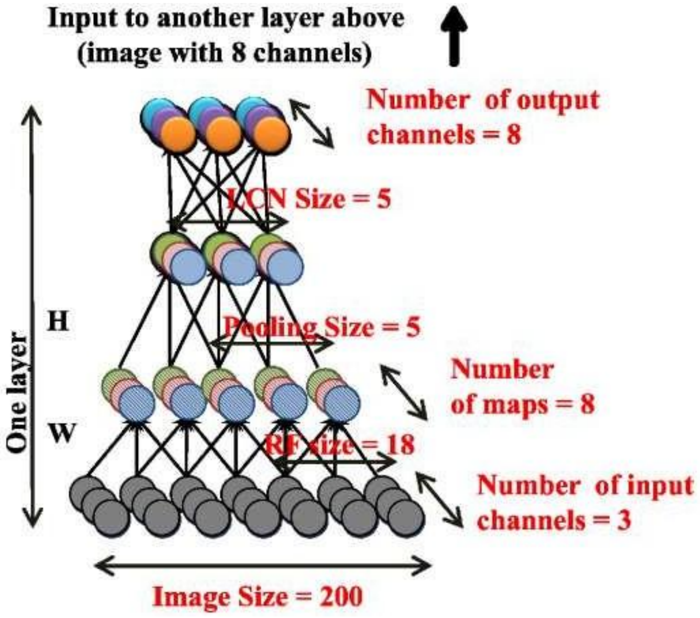
- In any directed network of computational elements with input source nodes and output sink nodes, “depth” is the length of the longest path from a source to a sink



- Left: Depth = 2.   Right: Depth = 3

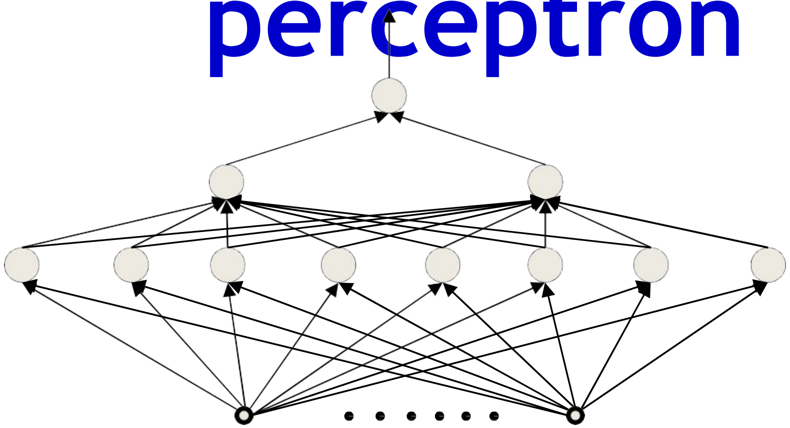
Deep Structures

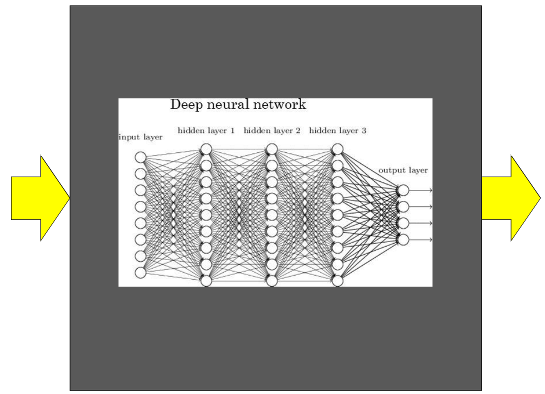
- *Layered* deep structure



- "Deep" = Depth greater than 2

The multi-layer perceptron





- Inputs are real or Boolean stimuli
- Outputs are real or Boolean values
  - Can have multiple outputs for a single input
- **What can this network compute?**
  - **What kinds of input/output relationships can it model?**

The MLP as a Boolean function

- How well do MLPs model Boolean functions?

How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

- ***A Boolean function is just a truth table***

How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$

- Expressed in disjunctive normal form (DNF)

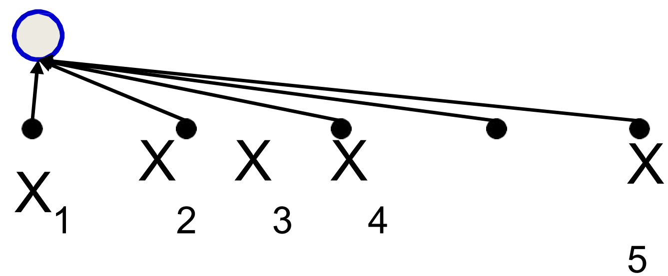
How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$



- Expressed in disjunctive normal form

How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

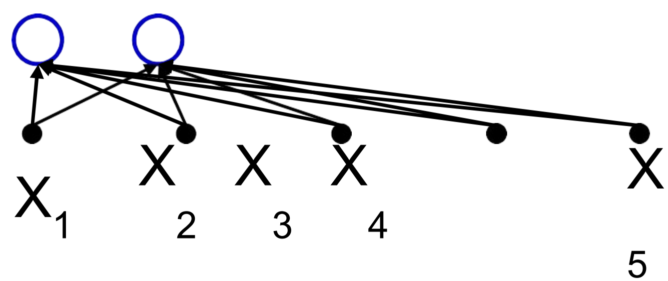
Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

$$Y - \bar{X}_1\bar{X}\ X_2X\ \bar{X}\ + \bar{X}_1X\ \bar{X}_2X\ X + \bar{X}_1X\ X_2\bar{X}\ \bar{X}\ +$$
$$X3\bar{X}\ \bar{X}4\bar{X}\ X\ + X3\bar{X}\ \bar{X}4X\ X\ + X3X\ \bar{X}4\bar{X}\ X$$

5 5 5

1 2 1 2 1 2
3 4 3 4 3 4
5 5 5



- Expressed in disjunctive normal form

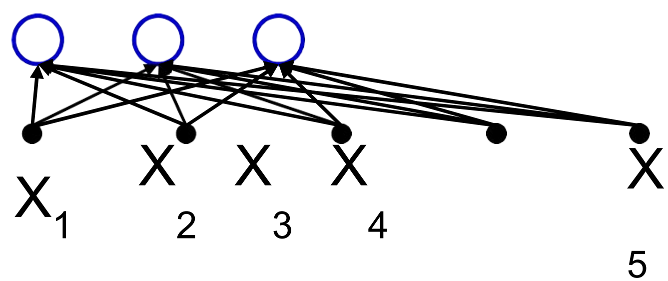
How many layers for a Boolean MLP?

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

Truth table shows *all* input combinations for which output is 1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 +$$



- Expressed in disjunctive normal form

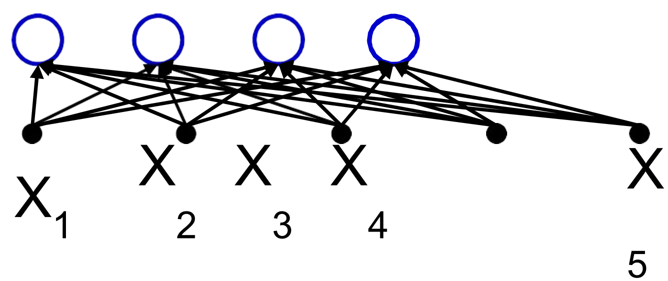
How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$



- Expressed in disjunctive normal form

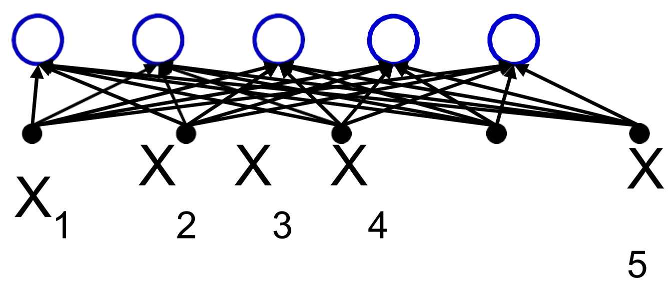
How many layers for a Boolean MLP?

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

Truth table shows *all* input combinations for which output is 1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$



- Expressed in disjunctive normal form

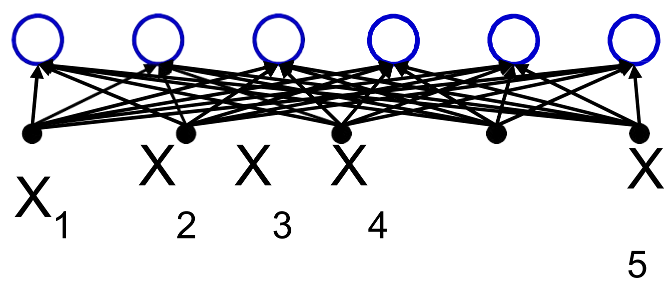
How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$



- Expressed in disjunctive normal form

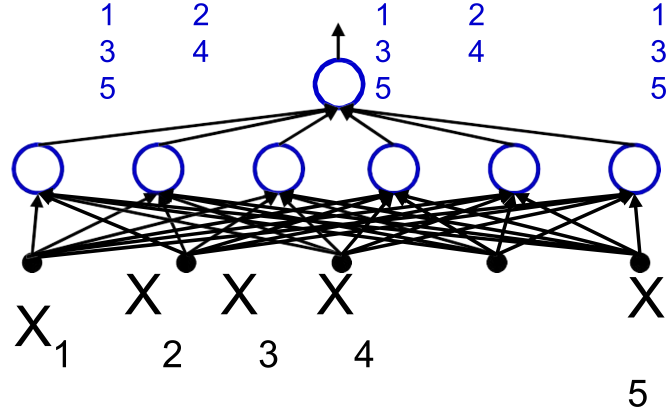
How many layers for a Boolean MLP?

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

Truth table shows *all* input combinations for which output is 1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$



- Expressed in disjunctive normal form

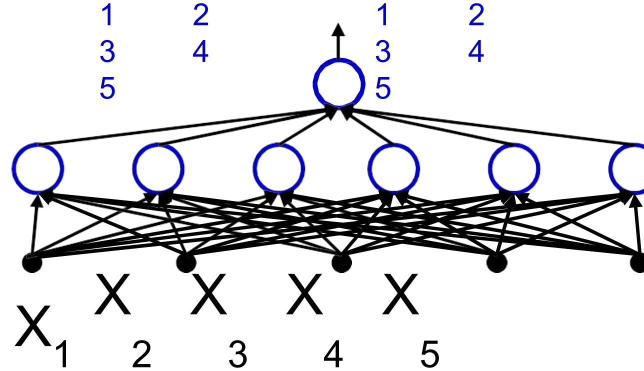
How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

Truth Table

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$



- *Any truth table can be expressed in this manner!*
- **A one-hidden-layer MLP is a Universal Boolean Function**

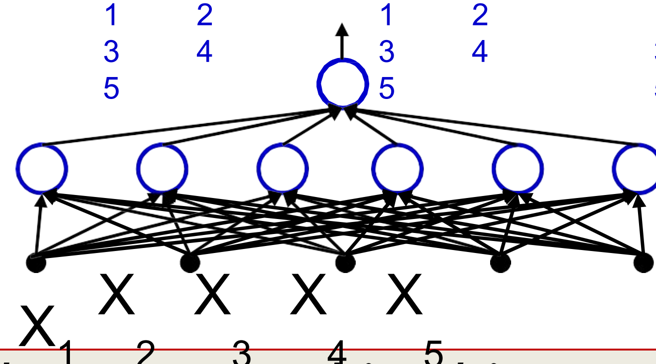
How many layers for a Boolean MLP?

Truth table shows *all* input combinations for which output is 1

Truth

X_1	X_2	X_3	X_4	X_5	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

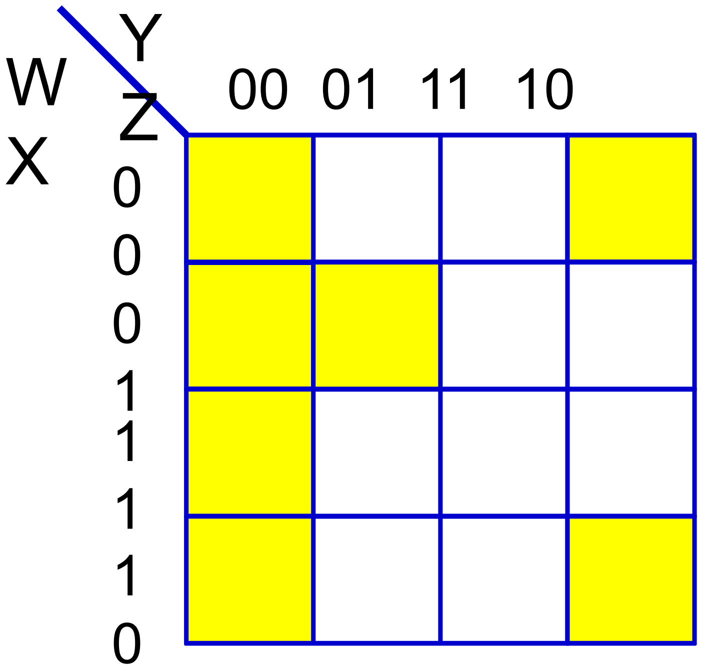
$$Y = \bar{X}_1\bar{X}_2X_3X_4\bar{X}_5 + \bar{X}_1X_2\bar{X}_3X_4X_5 + \bar{X}_1X_2X_3\bar{X}_4\bar{X}_5 + X_1\bar{X}_2\bar{X}_3\bar{X}_4X_5 + X_1\bar{X}_2X_3X_4X_5 + X_1X_2\bar{X}_3\bar{X}_4X_5$$



- *Any truth table can be expressed in this manner!*
- **A one-hidden-layer MLP is a Universal Boolean Function**

But what is the largest number of perceptrons required in the single hidden layer for an N-input-variable function?

Reducing a Boolean Function



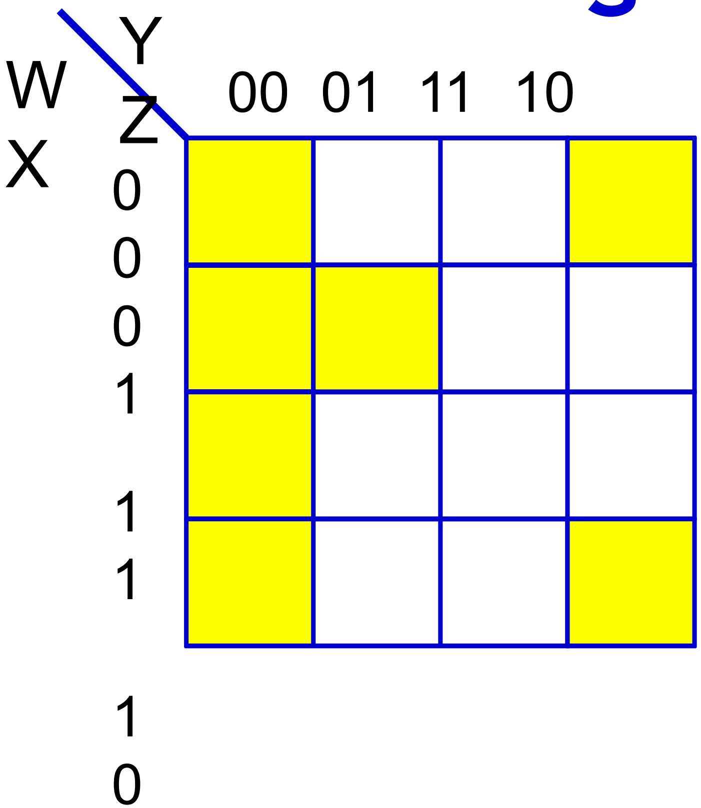
This is a "Karnaugh Map"

It represents a truth table as a grid
Filled boxes represent input combinations for which output is 1; blank boxes have output 0

Adjacent boxes can be "grouped" to reduce the complexity of the DNF formula for the table

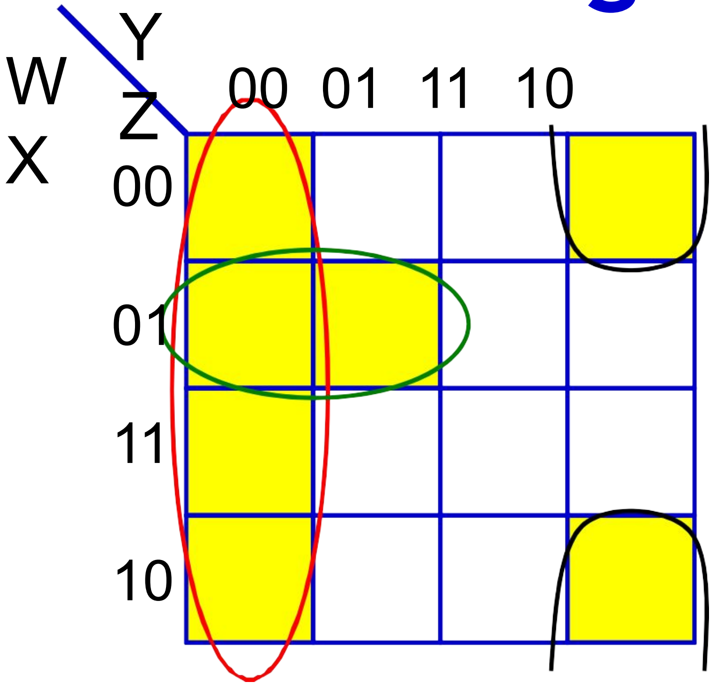
- DNF form:
  - Find groups
  - Express as reduced DNF

Reducing a Boolean Function



Basic DNF formula will require 7 terms

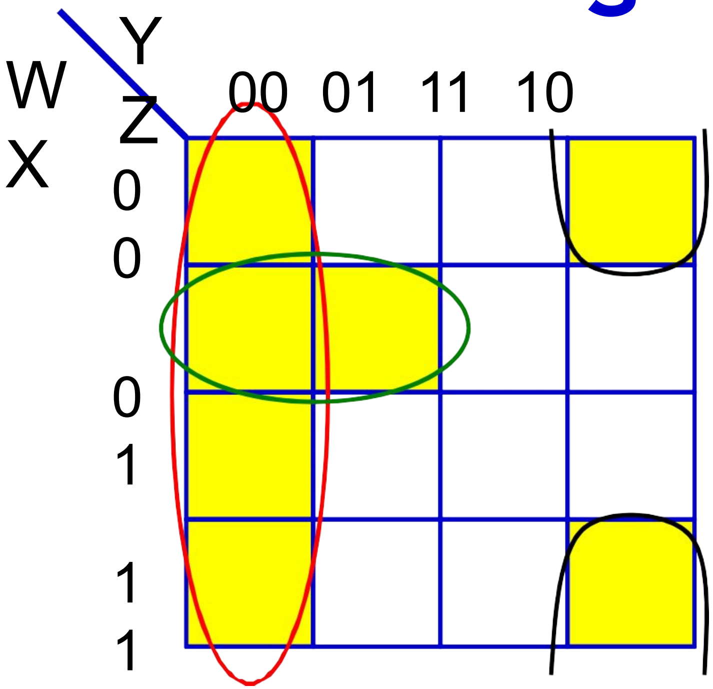
Reducing a Boolean Function



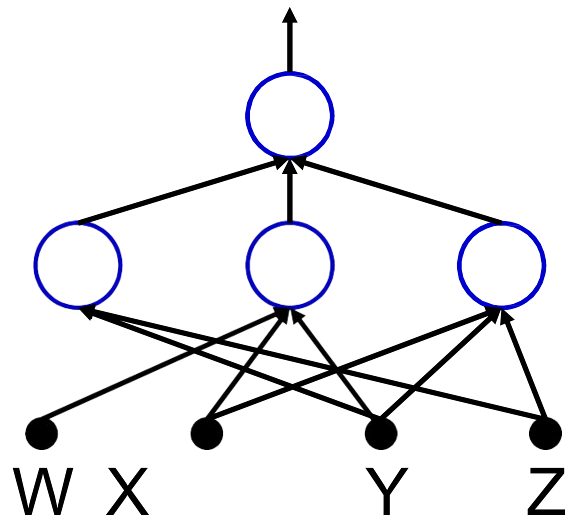
$O = \bar{Y}\bar{Z} + \bar{W}X\bar{Y} + \bar{X}Y\bar{Z}$

- *Reduced* DNF form:
  - Find groups
  - Express as reduced DNF

Reducing a Boolean Function

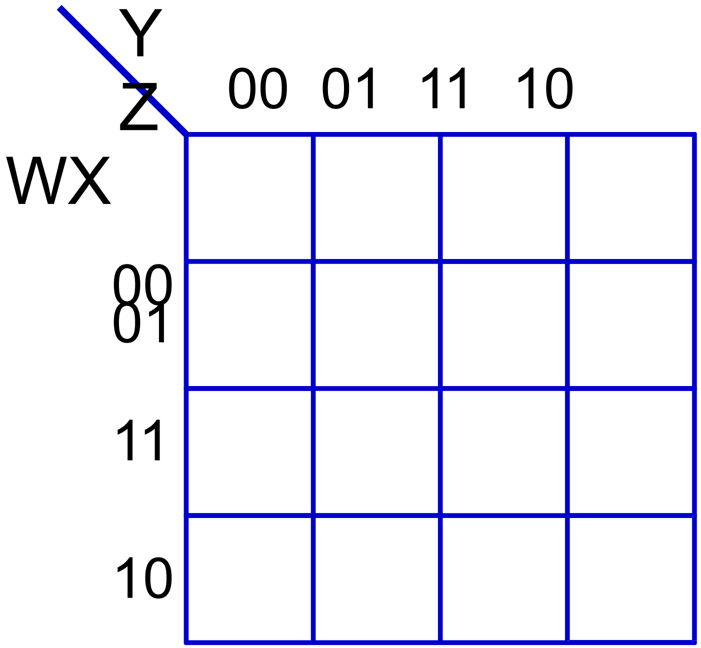


$$O = \bar{Y}\bar{Z} + \bar{W}X\bar{Y} + \bar{X}Y\bar{Z}$$



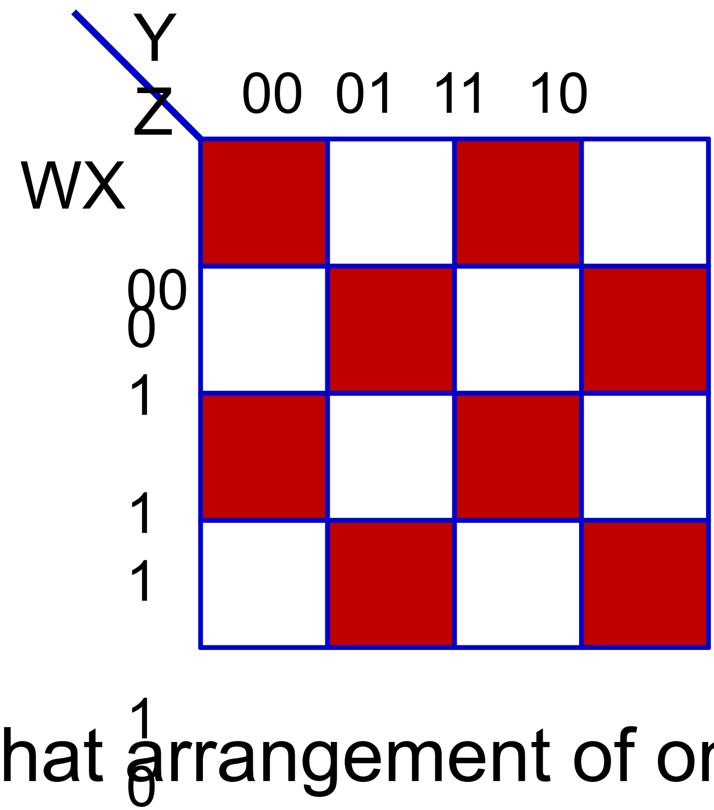
- *Reduced* DNF form:
  - Find groups
  - Express as *reduced* DNF
  - Boolean network for this function needs only 3 hidden units
    - Reduction of the DNF reduces the size of the one-hidden-layer network

Largest irreducible DNF?



- What arrangement of ones and zeros simply cannot be reduced further?

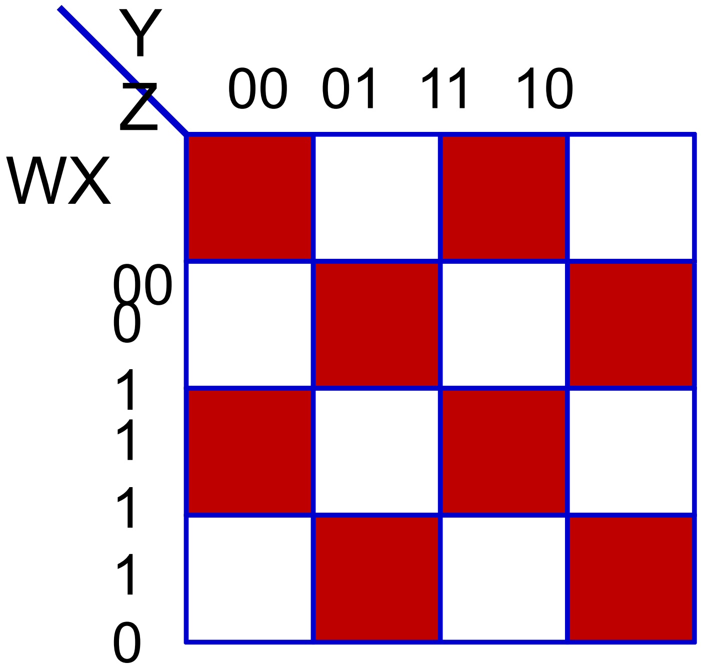
Largest irreducible DNF?



Red=0,
white=1

- What arrangement of ones and zeros simply cannot be reduced further?

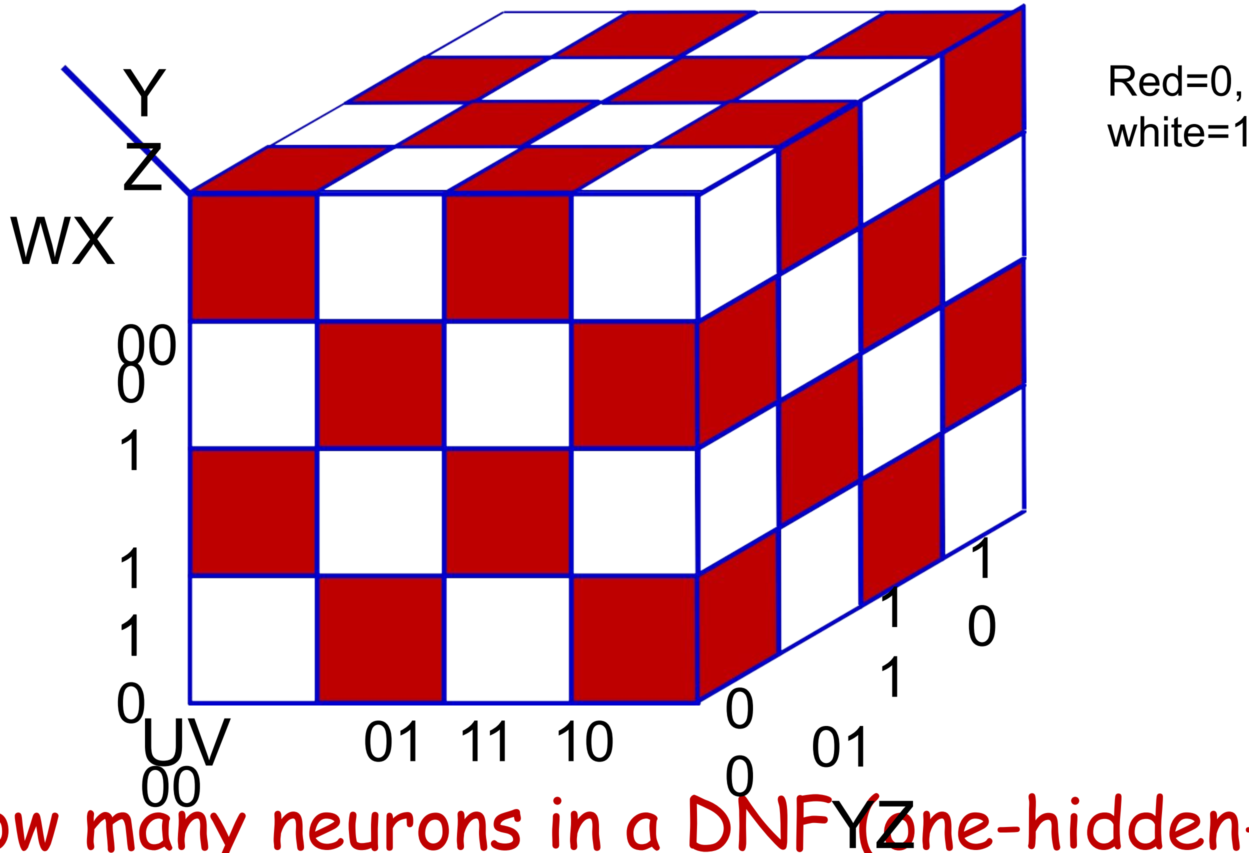
Largest irreducible DNF?



How many neurons in a DNF (one-hidden-layer) MLP for this Boolean function?

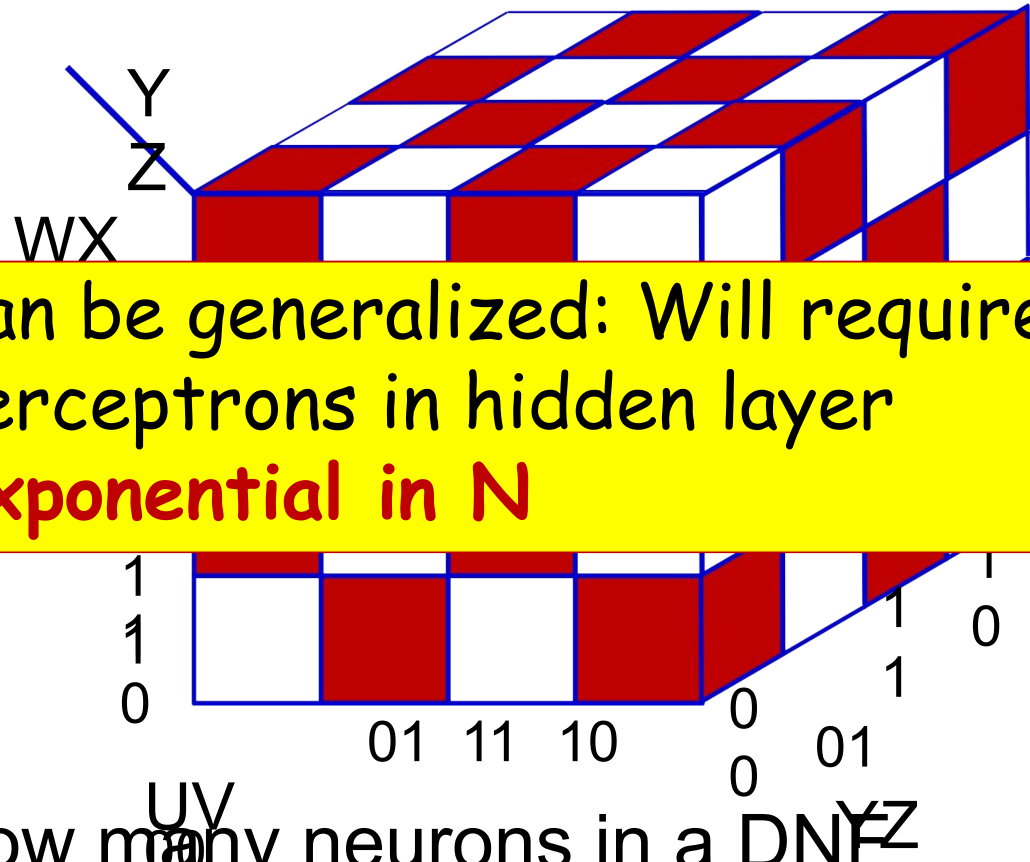
- What arrangement of ones and zeros simply cannot be reduced further?

Width of a single-layer Boolean MLP



- How many neurons in a DNF (one-hidden-layer) MLP for this Boolean function of 6 variables?

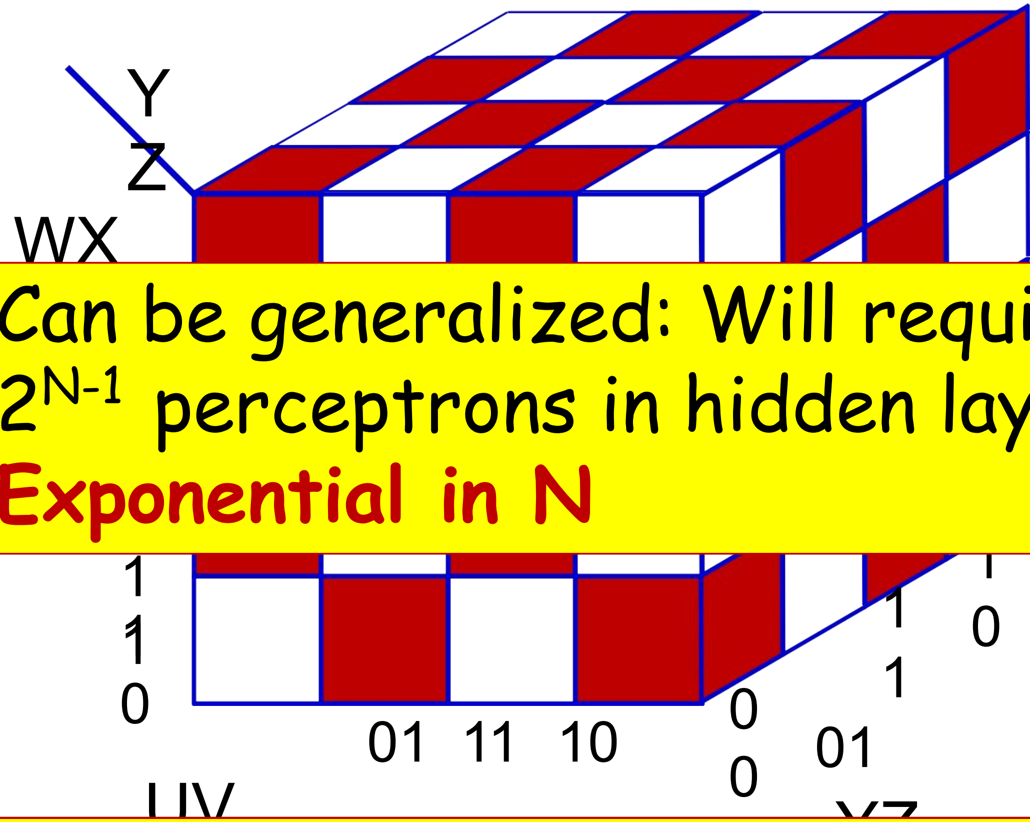
Width of a single-layer Boolean MLP



Can be generalized: Will require 2^{N-1} perceptrons in hidden layer
Exponential in N

- How many neurons in a DNF (one-hidden-layer) MLP for this Boolean function

Width of a single-layer Boolean MLP

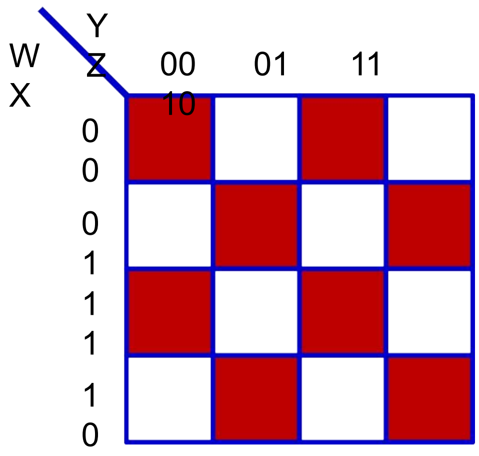


Can be generalized: Will require 2^{N-1} perceptrons in hidden layer
Exponential in N

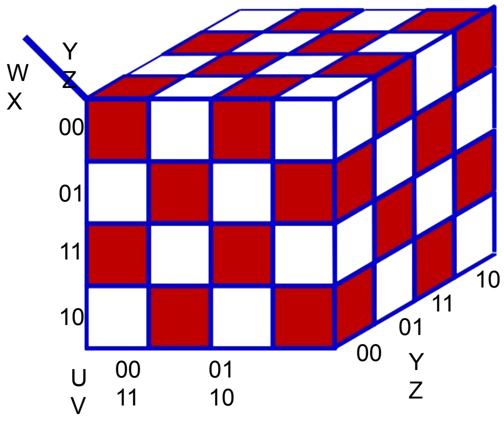
How many units if we use *multiple* *layers?*

(one-hidden layer) MLP for this Boolean function

Width of a deep MLP

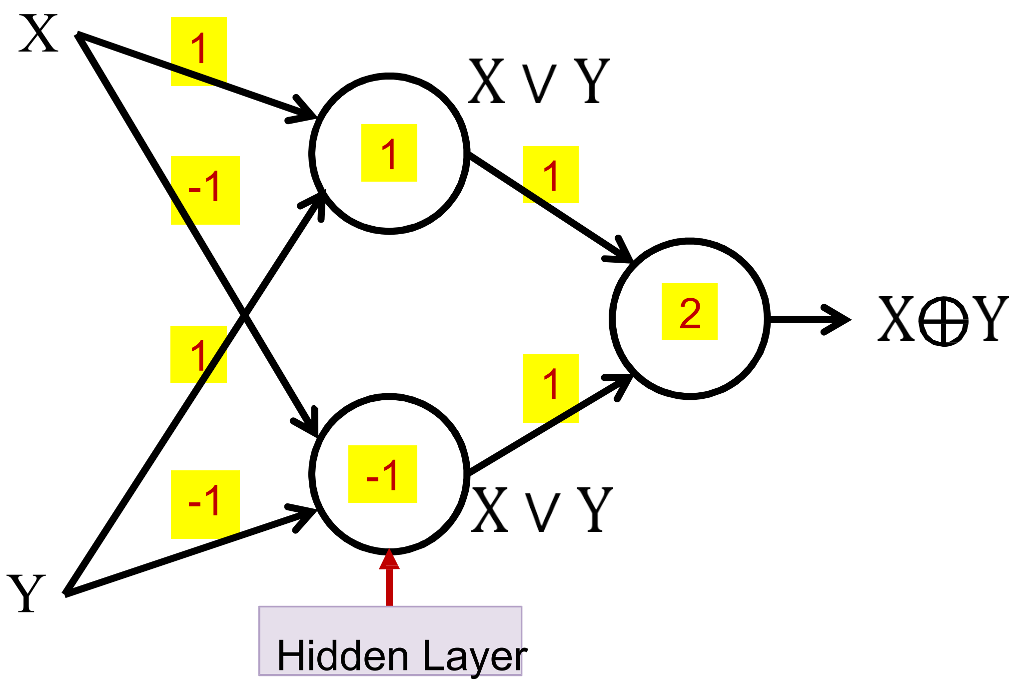


$$O = W \oplus X \oplus Y \oplus Z$$



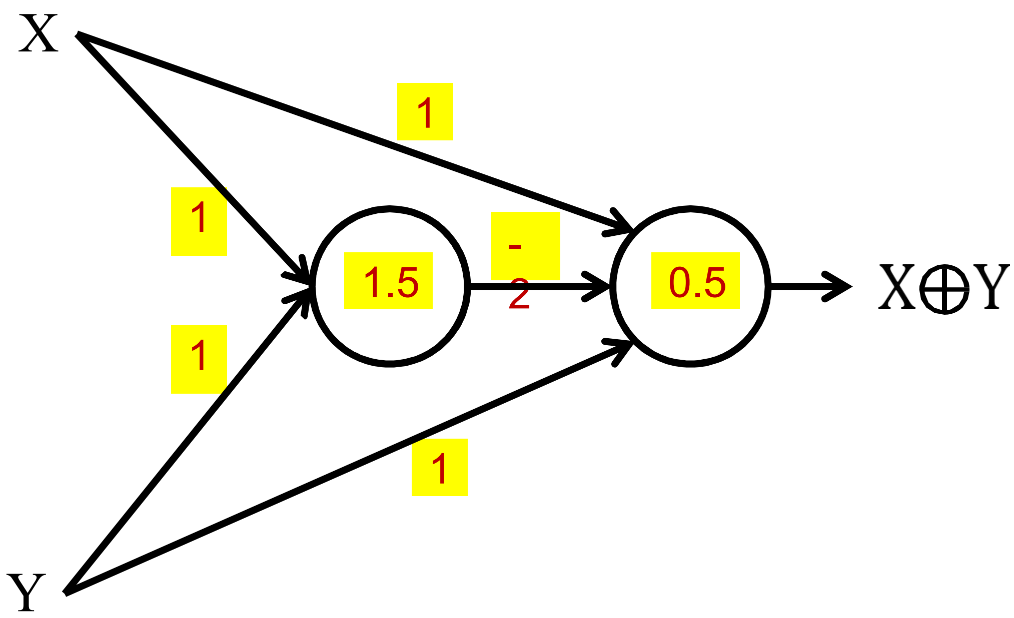
$$O = U \oplus V \oplus W \oplus X \oplus Y \oplus Z$$

Multi-layer perceptron XOR



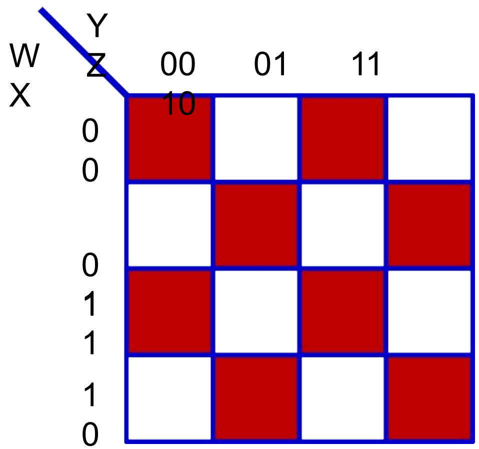
- An XOR takes three perceptrons

Multi-layer perceptron XOR

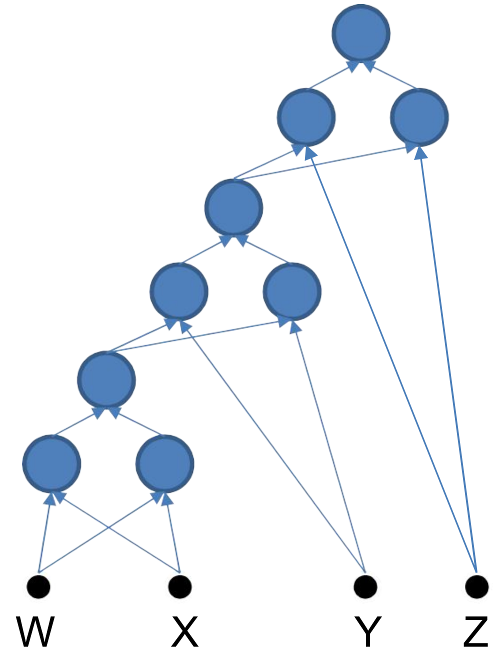


- With 2 neurons
  - 5 weights and two thresholds

Width of a deep MLP



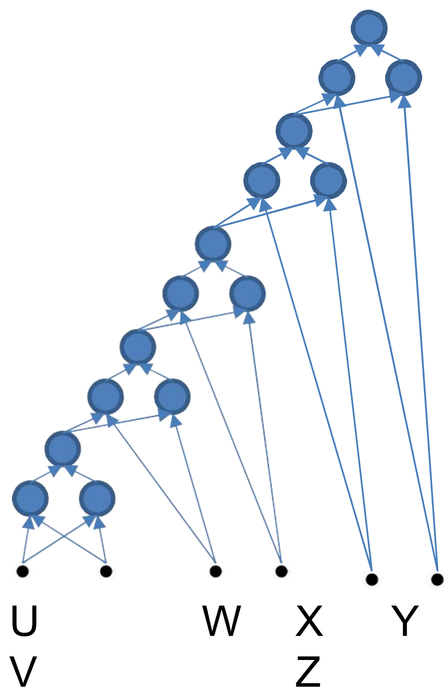
$$O = W \oplus X \oplus Y \oplus Z$$

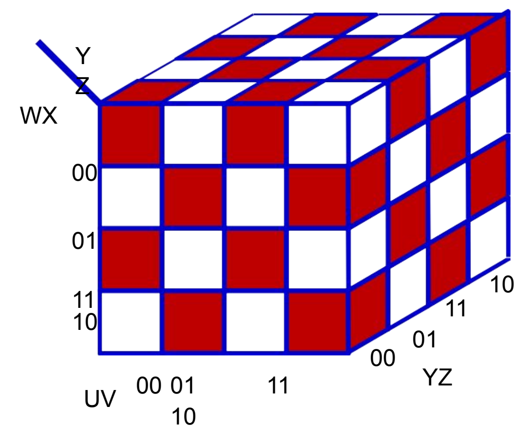


9 perceptrons

- An XOR needs 3 perceptrons
- This network will require 3x3 = 9 perceptrons

Width of a deep MLP



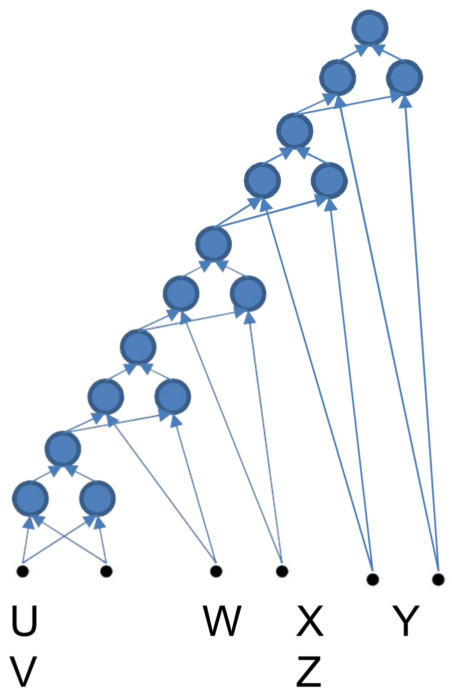


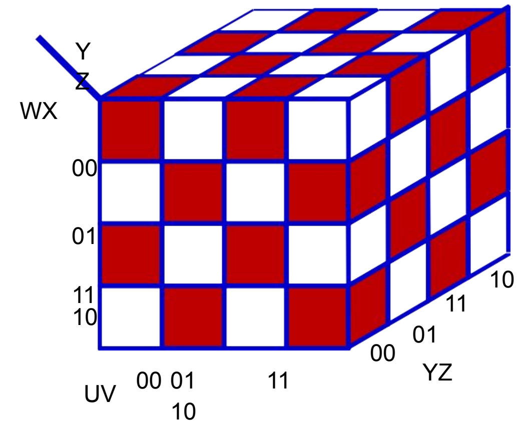
$$O = U \oplus V \oplus W \oplus X \oplus Y \oplus Z$$

15 perceptrons

- An XOR needs 3 perceptrons
- This network will require 3x5 = 15 perceptrons

Width of a deep MLP



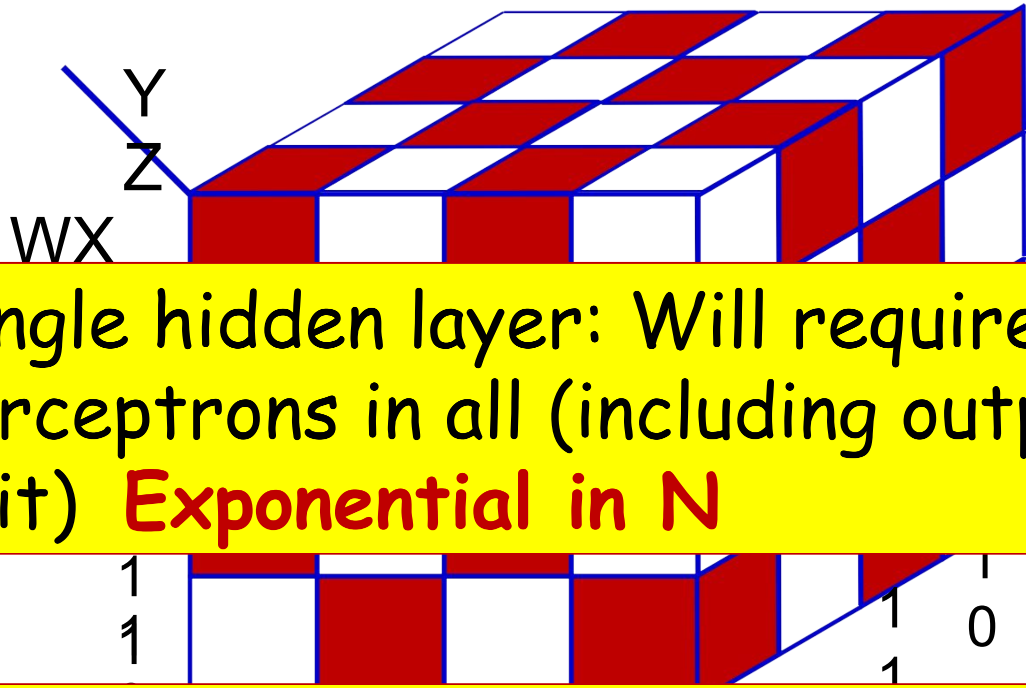


$$O = U \oplus V \oplus W \oplus X \oplus Y \oplus Z$$

More generally, the XOR of N variables will require 3(N-1) perceptrons!!

- An XOR needs 3 perceptrons
- This network will require 3x5 = 15 perceptrons

Width of a single-layer Boolean MLP



Single hidden layer: Will require $2^{N-1} + 1$ perceptrons in all (including output unit) **Exponential in N**

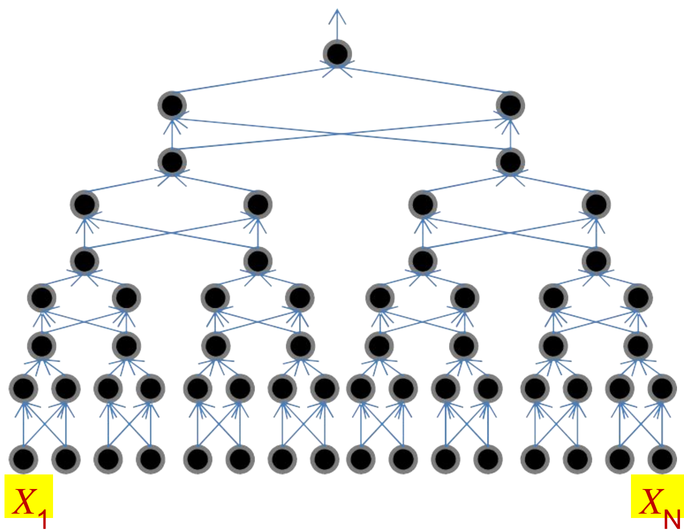
Will require 3(N-1) perceptrons in a deep network

Linear in N!!!

Can be arranged in only $2\log_2(N)$ layers

Boolean function

A better representation



$$O = X_1 \oplus X_2 \oplus \cdots \oplus X_N$$

- Only $2\log_2 N$ layers
  - By pairing terms
  - 2 layers per XOR

$$O = (((((X_1 \oplus X_2) \oplus (X_3 \oplus X_4)) \oplus ((X_5 \oplus X_6) \oplus (X_7 \oplus X_8))) \oplus (((\cdots$$

Recap: The need for depth

- *Deep* Boolean MLPs that scale *linearly* with the number of inputs …
- … can become exponentially large if recast using only one layer

- It gets worse..

Network size: summary

- An MLP is a universal Boolean function

- But can represent a given function only if
  - It is sufficiently wide
  - It is sufficiently deep
  - Depth can be traded off for (sometimes) exponential growth of the width of the network

- Optimal width and depth depend on the number of variables and the complexity of the Boolean function
  - Complexity: *minimal* number of terms in DNF formula to represent it

Story so far

- Multi-layer perceptrons are *Universal Boolean Machines*
  - Even a network with a *single* hidden layer is a universal Boolean machine

- Multi-layer perceptrons are *Universal Classification Functions*
  - Even a network with a single hidden layer is a universal classifier

- But a single-layer network may require an exponentially large number of perceptrons than a deep one

- Deeper networks may require far fewer neurons than shallower networks to express the same function
  - Could be *exponentially* smaller
  - Deeper networks are more *expressive*