JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

EN.601.482/682 Deep Learning

Introduction to and History of Neural Networks

Mathias Unberath, PhD
Assistant Professor
Dept of Computer Science
Johns Hopkins University

Organizational

- Homework 2 due soon (start early!)
- Homework 3 released then

- Final projects
  - Groups of 4
  - Primary goal
    - Demonstrate your understanding of solving problems using DL
    - Demonstrate reasoning behind design choices (e.g., through ablations)
  - Secondar goal
    - Have fun
    - Get super duper creative (after you checked the boxes for grading)
  - Requirements
    - Feasibility! If you do not have a data set, it’s not a suitable project!

Reminder

- The bias variance tradeoff

$$L(W) = \underbrace{(E[\hat{y}] - y)^2}_{\text{Bias}^2} + \underbrace{E[(\hat{y} - E[\hat{y}])^2]}_{\text{Variance}} + \sigma$$

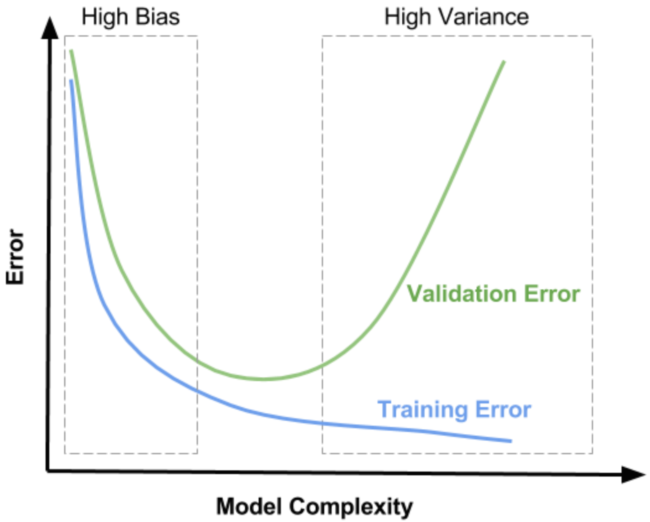
- Optimization:
  - Evaluate gradient of loss L at current estimate W

    $$\nabla L(\mathbf{W}) = \left(\frac{\partial L}{\partial W_1} \quad \frac{\partial L}{\partial W_2} \quad \cdots \quad \frac{\partial L}{\partial W_n}\right)$$

  - Update W in the direction of steepest descent

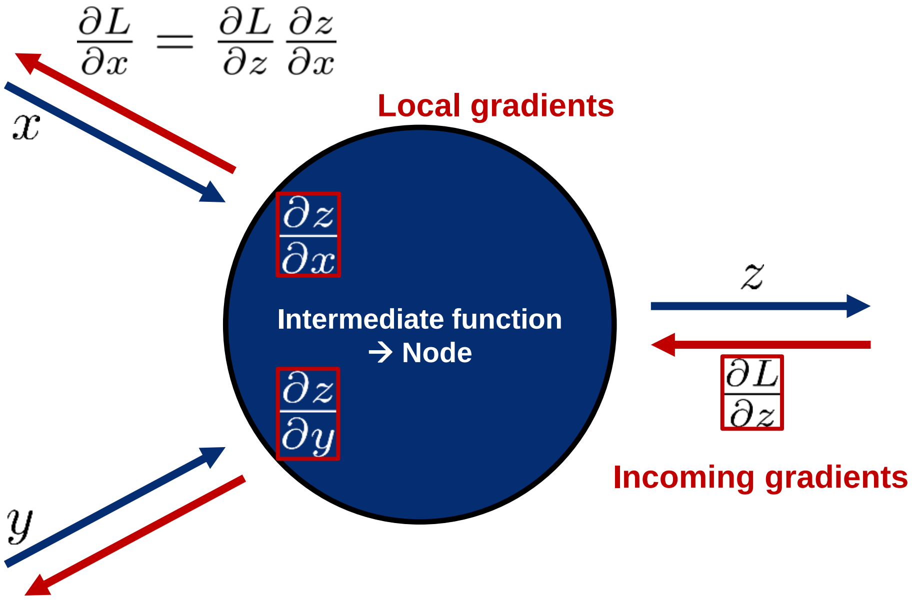
    $$\mathbf{W}' = \mathbf{W} - \lambda \nabla L(\mathbf{W})$$

  - When to stop?



Reminder

- Computational graphs



- Compute derivatives anywhere w.r.t. anything via backpropagation
- Vector / matrix backprop gets high-dimensional fast!

$$\boxed{\frac{\partial f}{\partial x_i} = \frac{\partial f}{\partial (q_1, \ldots, q_m)} \frac{\partial (q_1, \ldots, q_m)}{\partial x_i} = \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial x_i}}$$

Reminder

Goal for today:
Understand cognitive foundation of neural networks and their history.

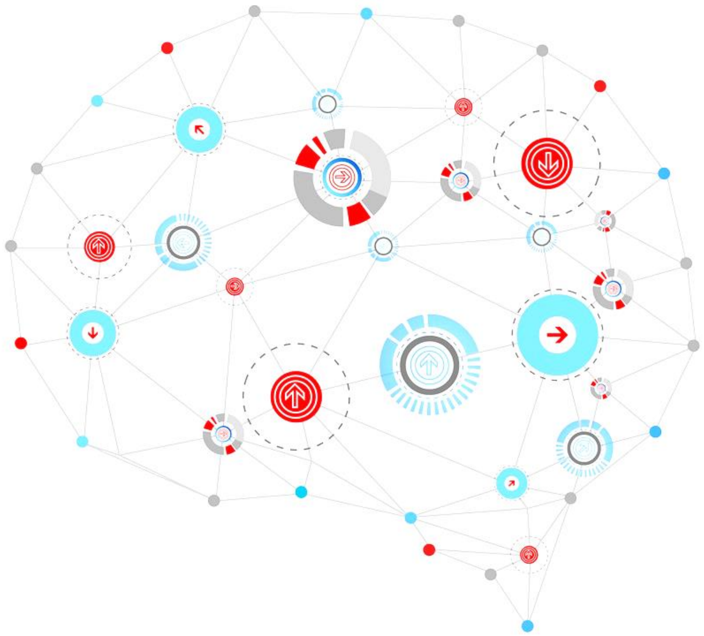
Today’s Lecture

History of and

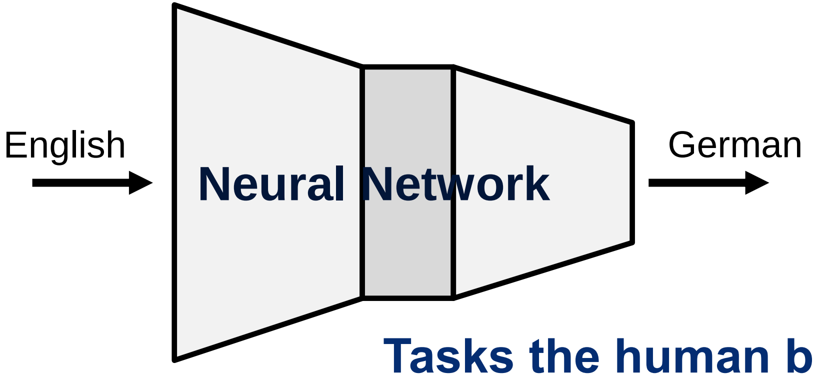
Introduction to Neural Networks

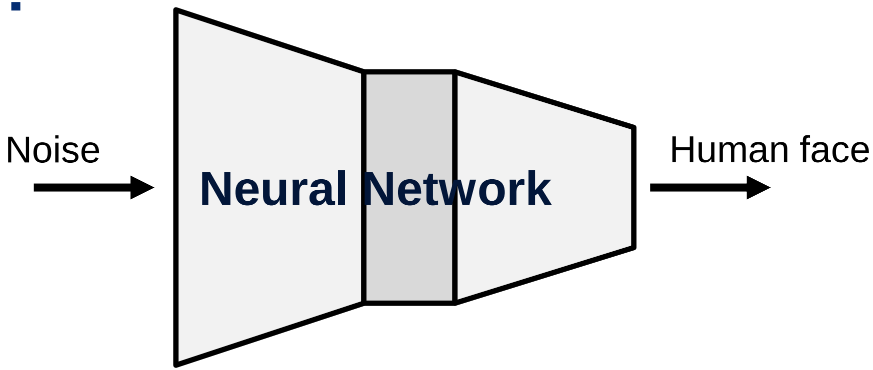
Multi-layer Perceptrons as

Universal Boolean Functions

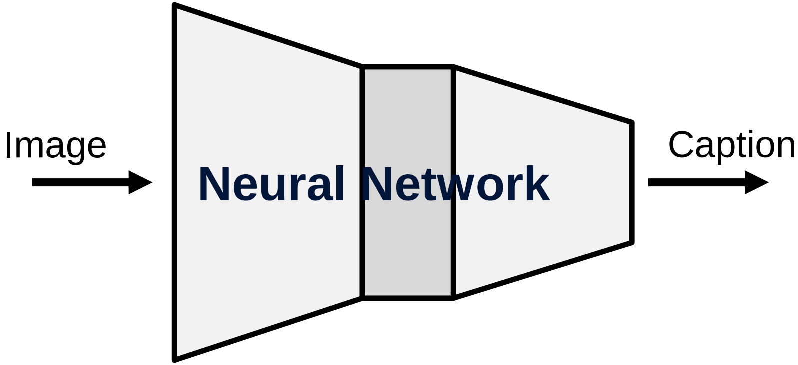


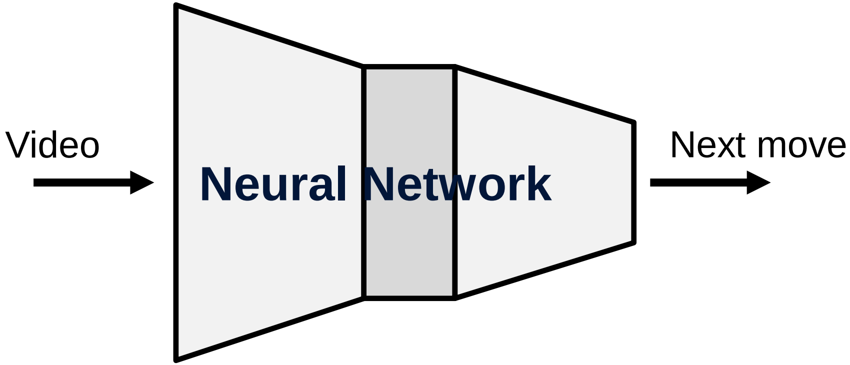
What are Neural Networks?





Tasks the human brain does easily “on the go”



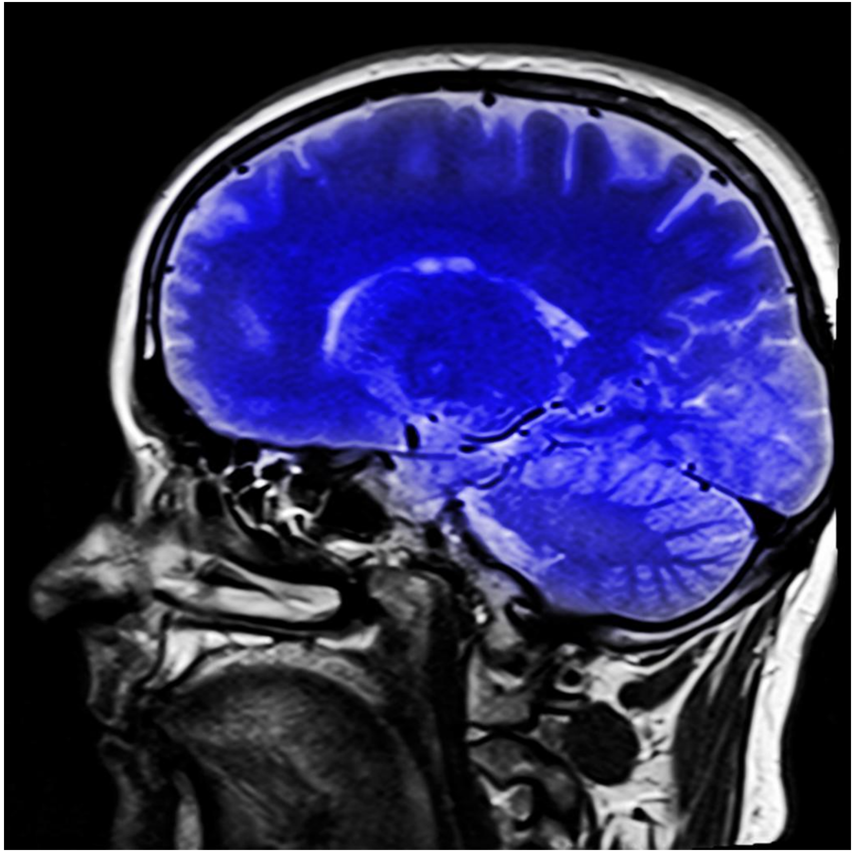


It Begins With…

With this!

All of these tasks are human tasks,
that are performed by the human brain!

→ We have to understand the human brain.



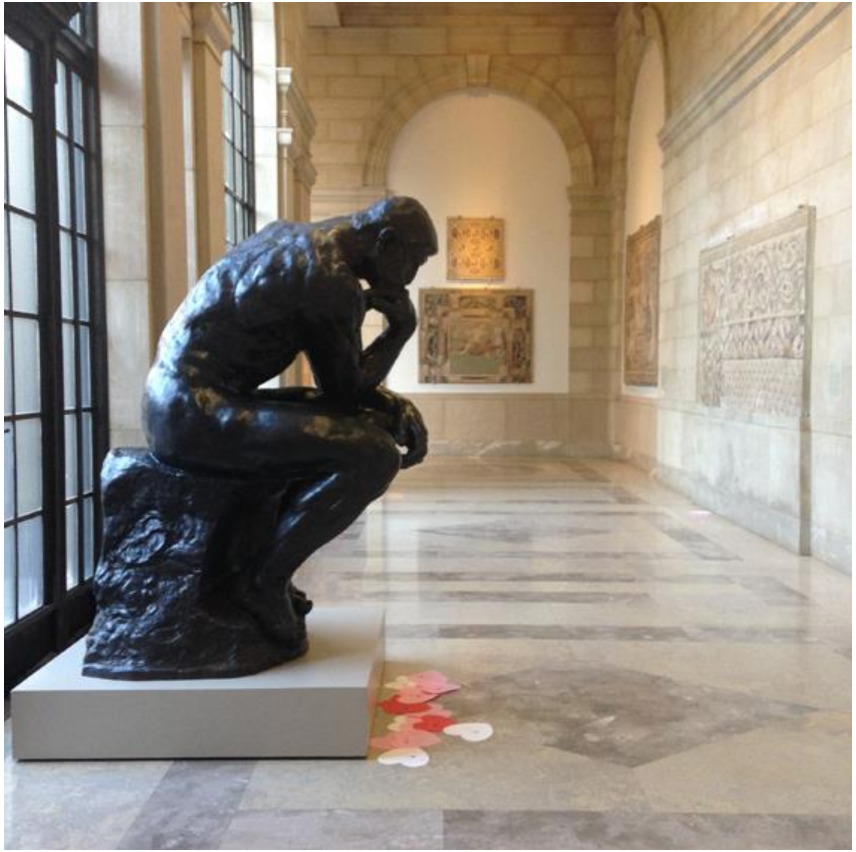
It Begins With…

Or even earlier with this!

With cognition:

What does it mean to think?

What does it mean to cogitate?



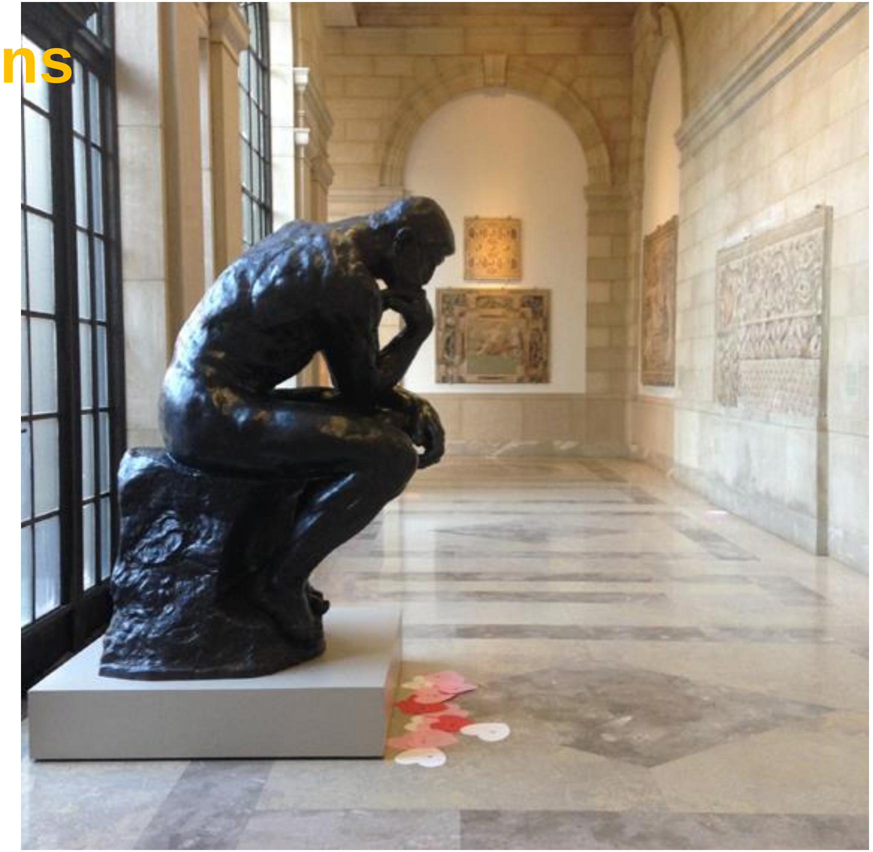
The Thinker - Auguste Rodin

The Magical Capacity of Humans

Humans can

- Learn
- Solve problems
- Recognize patterns
- Create
- …

Worthy of emulation **but**,
how do humans work?



The Thinker - Auguste Rodin

Cognition is a Complex Process

“If the brain was simple enough to be understood –
we would be too simple to understand it!”

– Marvin Minsky,

American Cognitive Scientist concerned with AI

Early Models of Human Cognition

Associationism: (one of the oldest theories of thought)
→ Humans learn through associations

400 BC – 1900 AD: Plato, David Hume, Ivan Pavlov

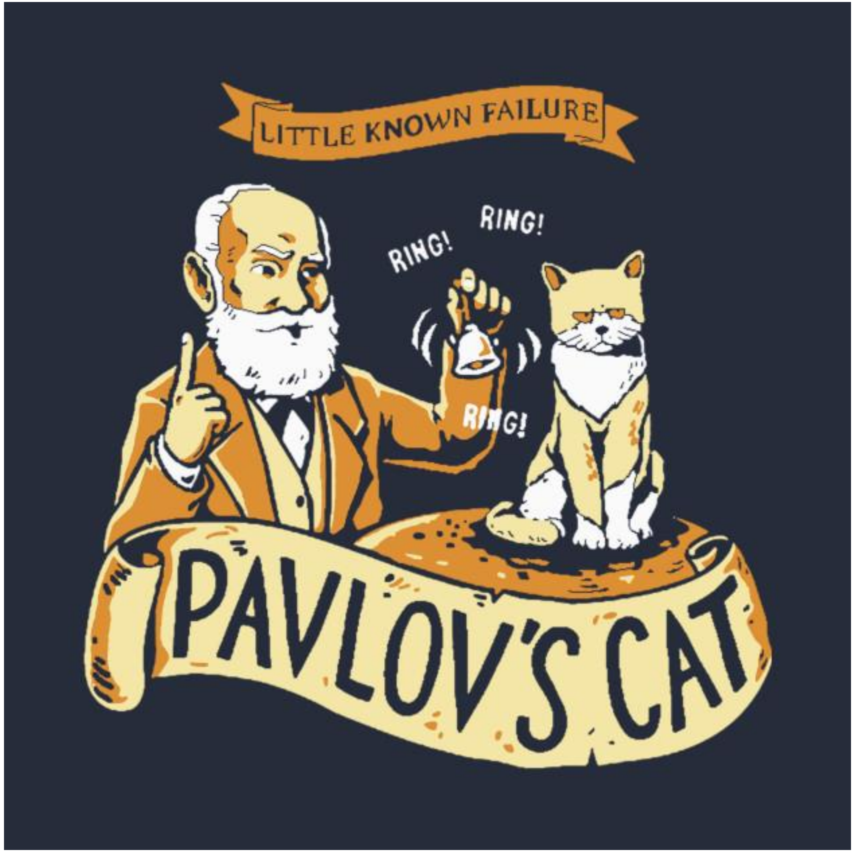
Q: Who has not heard about the Pavlovian reflex?

Associations

Lightning is generally followed by thunder

- Ergo: There’s a bold of lightning
  → We’re going to hear thunder!

- Ergo: We just heard thunder
  → Did someone get hit by lightning?



Machine learning algorithms even today still try to learn associations.

But:

Where are associations stored? … and how are they stored?

A quick

A Quick Aside

What is the difference between an associations and a cause?

A Quick Aside

What is the difference between an associations/correlation and a cause?

Example:

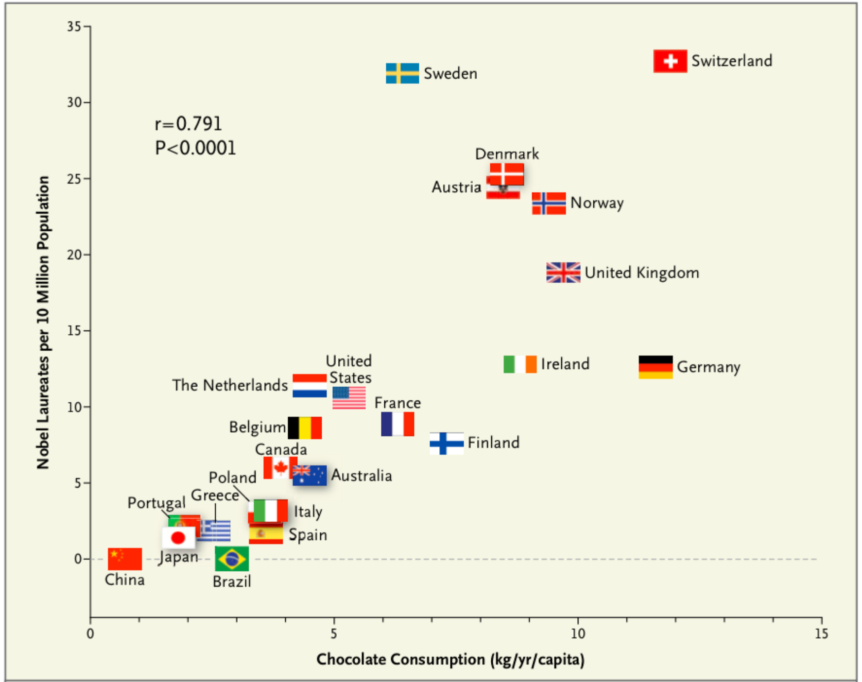


Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

A Quick Aside

What is the difference between an associations/correlation and a cause?

Example: Clearly, correlation between chocolate consumption and #nobel prizes

But is this a cause? No, it’s a correlation/association.

Why?

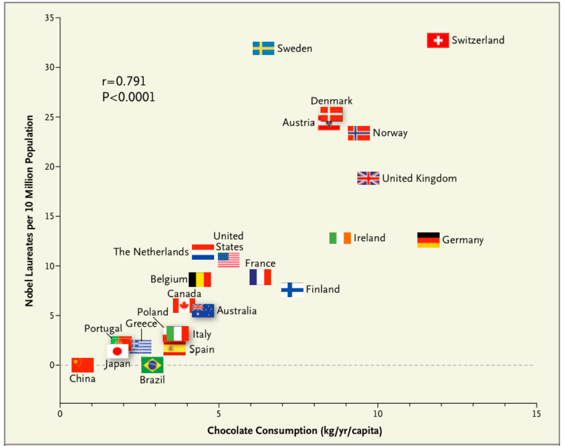


Figure 1. Correlation between Countries’ Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

A Quick Aside

What is the difference between an associations/correlation and a cause?

Example: Clearly, correlation between chocolate consumption and #nobel prizes

But is this a cause? No, it’s a correlation/association.

Why?

Confounding.

→ Most of ML does not differentiate cause/association.

Country

Chocolate consumption

Nobel prizes

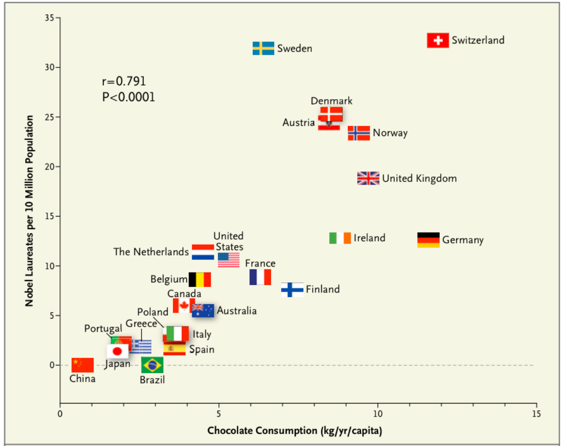


Figure 1. Correlation between Countries’ Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

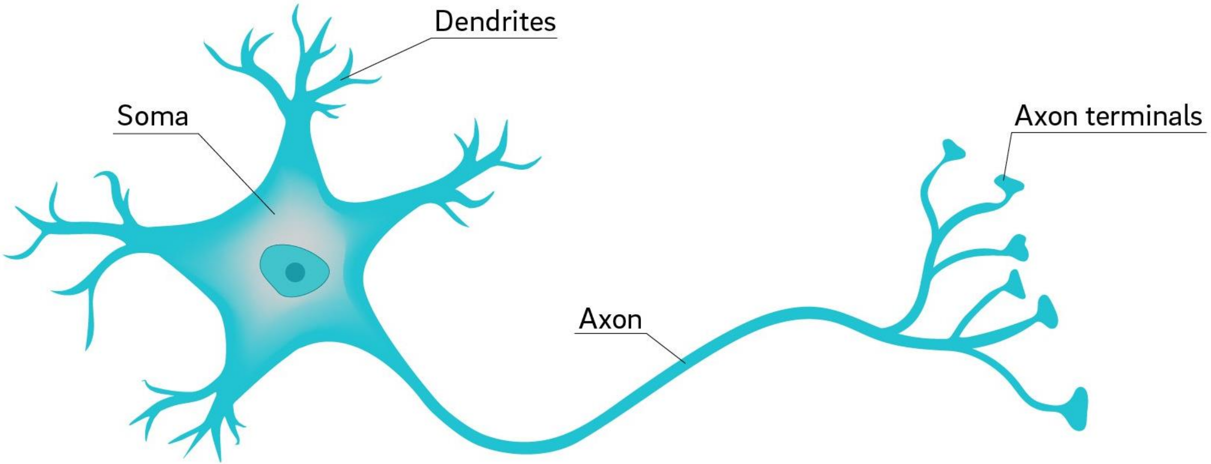
Machine learning algorithms even today still try to learn associations.

But:

Where are associations stored? … and how are they stored?

The Brain

Mid 1800s: The brain is a mass of interconnected tissue



Why mid 1800s: Microscopes of sufficient resolution.

→ The structure is known, so how does it store associations?

Connectionism

“The information is in the **connections**.”

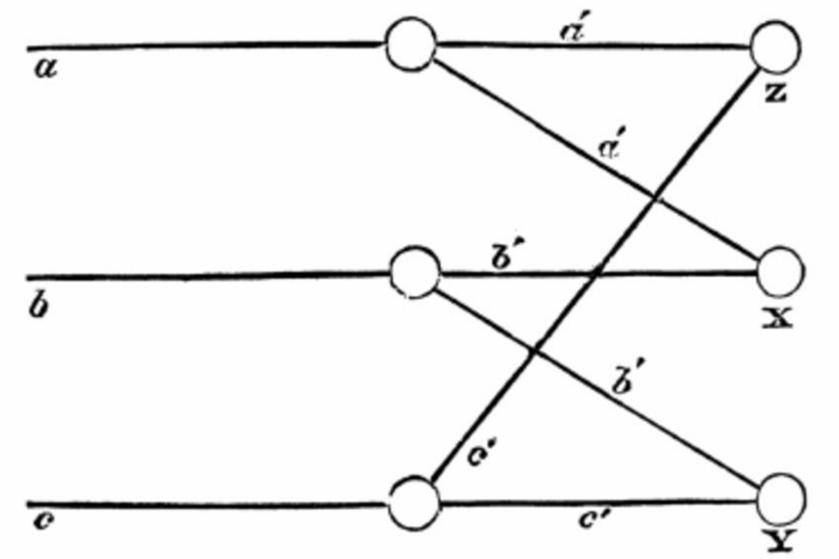
– Alexander Bain (1873):

Philosopher, mathematician, logician, linguist, professor

[Bain, A. (1873). Mind and Body the Theories of Their Relation. Henry S. King & Company.]

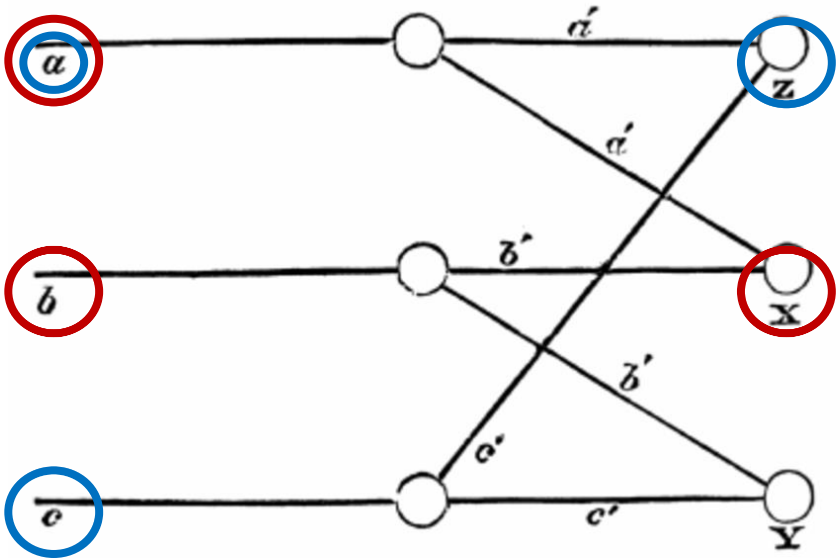
Bain’s Idea: Neural Groupings

- Neurons excite and stimulate one another
- Different combinations of inputs can result in different outputs
  → Trivial today but revolutionary back then!



Bain's Idea: Neural Groupings

- Neurons excite and stimulate one another
- Different combinations of inputs can result in different outputs
  → Trivial today but revolutionary back then!

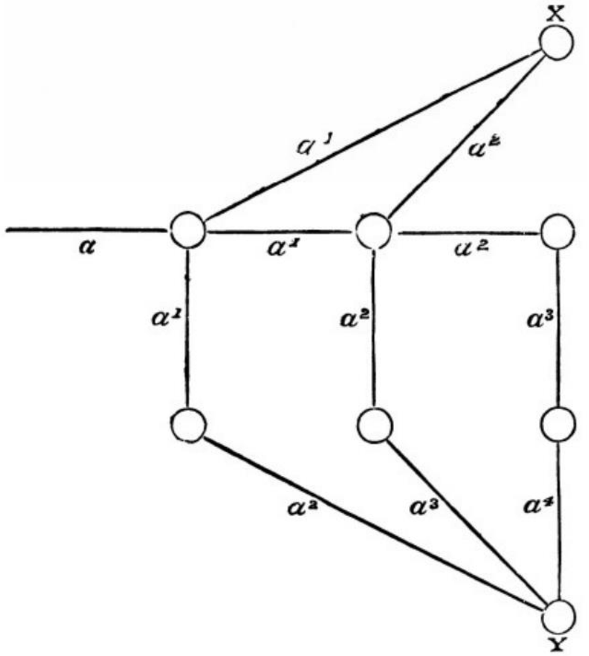


Bain's Idea: Neural Groupings

- Different intensities of activation lead to difference in X, Y

- If **a** is weak, only Y will fire
  - X is getting two copies of a
  - Y is getting three copies

- If **a** is strong, both X and Y will fire

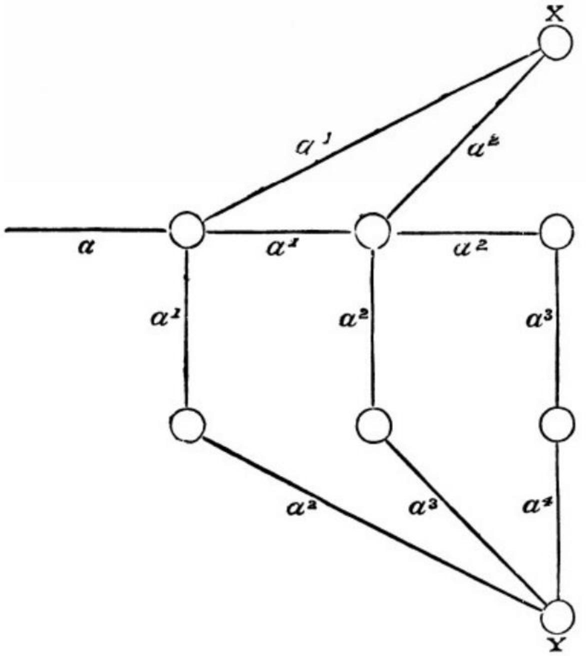


Bain's Idea: Neural Groupings

Bain even proposed a learning algorithm:

"When two impressions concur, or closely succeed one another, the nerve currents find some bridge or place of continuity, better or worse, according to the abundance of nerve matter available for the transition."

Today, this is known as Hebbian learning.



Bain's Doubts

"The fundamental cause of the trouble is that in the modern world the stupid are cocksure while the intelligent are full of doubt."

– Bertrand Russell

- Bain, 1873: To store 200,000 associations, one would need one million neurons and 5 billion connections
- Bain,1883: Had not taken into account "partially formed associations" and the number of neurons responsible for recall/learning
  → There is no way you can store 10 million neurons in a single head!
- By the end of his life (1903), recanted all his ideas!
  → Too complex

Bain’s Doubts

“The fundamental cause of the trouble is that in the modern world the stupid are cocksure while the intelligent are full of doubt.”

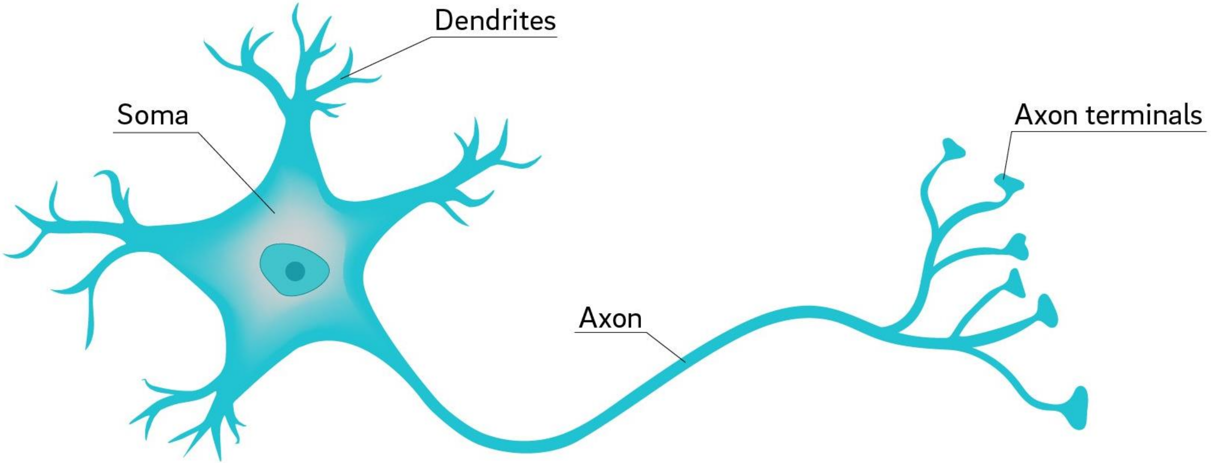
– Bertrand Russell

- Bain, 1873: To store 200,000 associations, one would need one million neurons and 5 billion connections
- Bain,1883: Had not taken into account “partially formed associations” and the number of neurons responsible for recall/learning
  → There is no way you can store 10 million neurons in a single head!
- By the end of his life (1903), recanted all his ideas!
  → Too complex

→ **Recall**: Human brain has ~ 10^{11} neurons, ~10^{14} synapses

Connectionism Continues

- The human brain is a connectionist machine

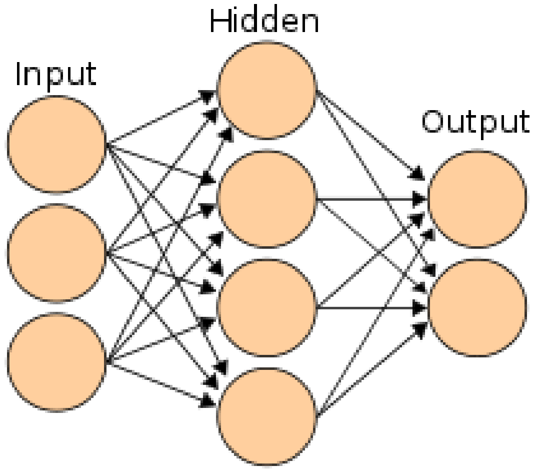


- Neurons connect to other neurons → Capacity depends on these connections.

- Connectionist machines emulate this structure

Connectionist Machines

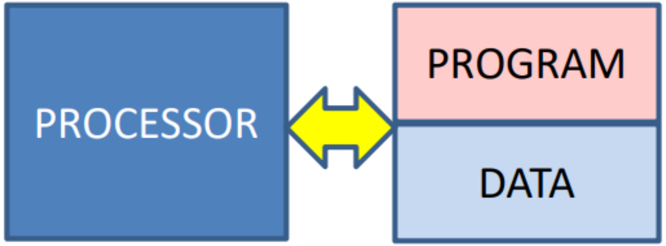
- Network of processing elements
- All knowledge is stored in the **connections** between these elements



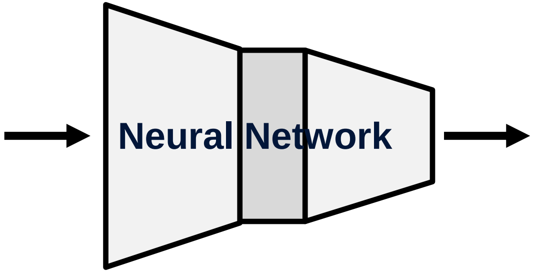
Connectionist Machines

- Neural networks are connectionist machines
- Van Neumann machines **are not**

Von Neumann machine



Memory



- A connectionist machine has many processing units
  → The program is the connection between these units
  → The connections may also define memory

Recap

- Neural networks originally began as computational models of the brain
  More generally, models of cognition

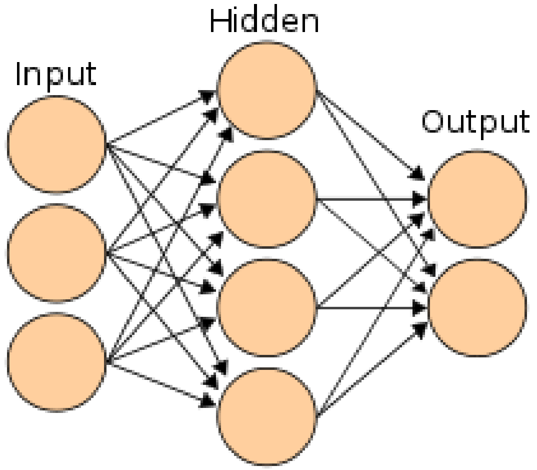
- Earliest model of cognition: Associationism

- More recent model: Connectionism
  → Neurons connect to neurons
  → Knowledge is encoded in these connections

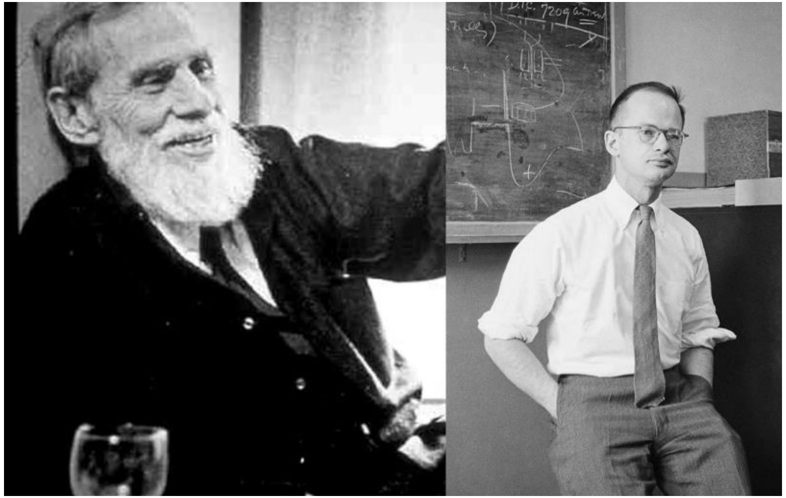
- Today's neural networks are connectionist machines

Connectionist Machines

- Network of processing elements
- All knowledge is stored in the **connections** between these elements
- What are these elements? Neurons!



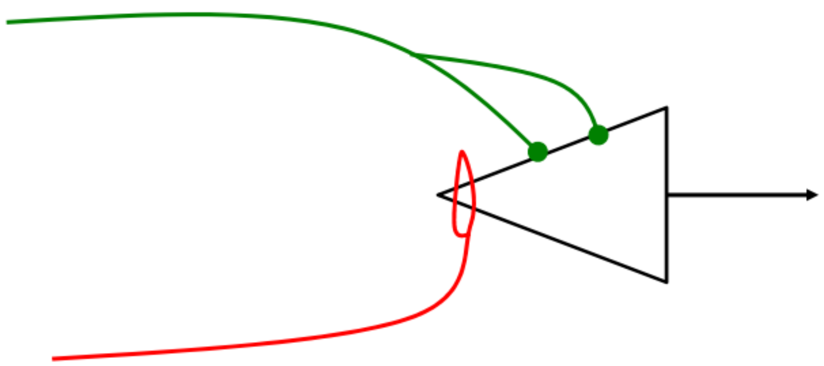
McCulloch and Pitts Model



Warren McCulloch: Neurophysician

Walter Pitts: Homeless wannabe logician

McCulloch and Pitts Model



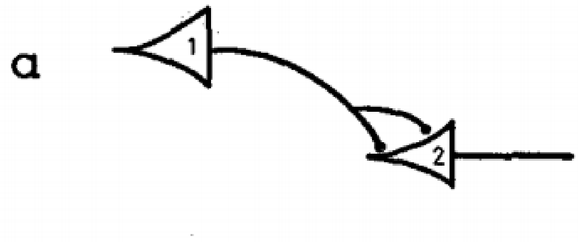
Every neuron requires **2 inputs to fire**

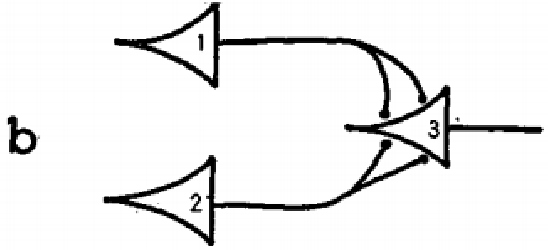
Excitatory synapse: Transmits input to the neuron

Inhibitory synapse: Forces output to zero

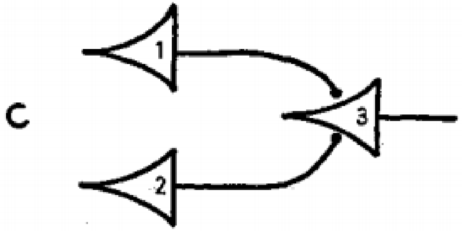
McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

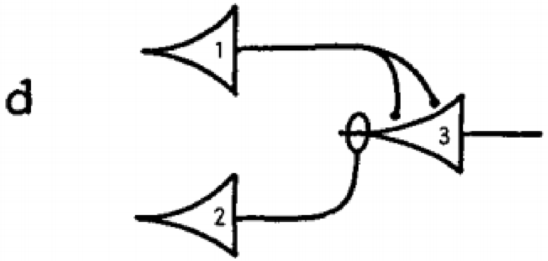
McCulloch and Pitts Model: Boolean Gates





Simple networks can
perform Boolean operations.
Which ones?





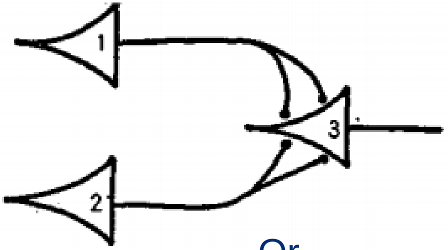
[McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.]

McCulloch and Pitts Model: Boolean Gates

a

“Synaptic delay”

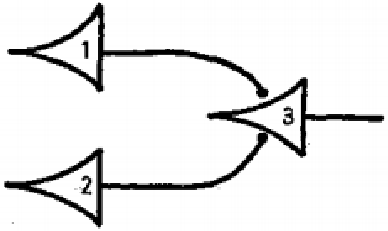
b



Or

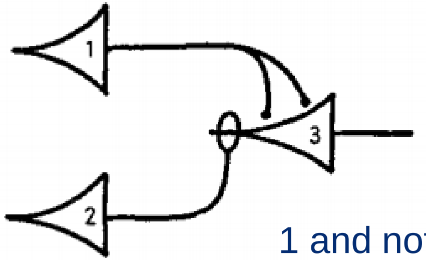
Simple networks can
perform Boolean operations.

c



And

d



1 and not 2

[McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.]

McCulloch and Pitts Model: Criticism

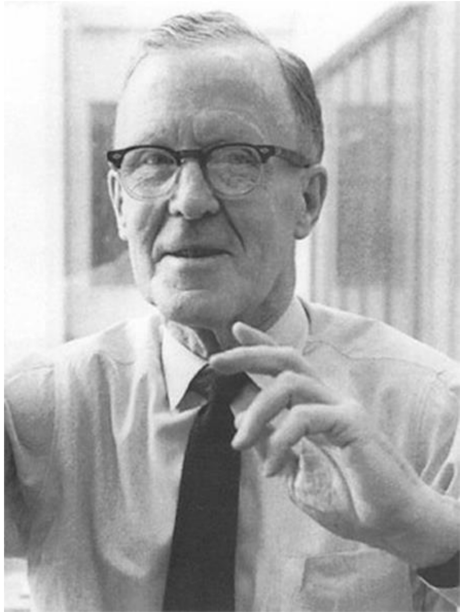
- Claimed that their networks
  - Should be able to compute a small class of functions
  - If tape is provided, networks can compute a richer class of functions
    - Will be equivalent to Turing machines
    - Claim that they are Turing complete
  - Did not prove any results themselves

- **No learning mechanism!**

Donald Hebb

Provides a learning mechanism:

“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.”



As A repeatedly excites B, its ability to excite B improves

→ *Neurons that fire together, wire together.*

→ But, what does it mean?

Hebb, D. O. (1949). The organization of behavior.

Hebbian Learning

- If neuron x_i repeatedly triggers x_j, the synaptic knob connecting both grows

- Mathematical model $w_{i,j} = w_{i,j} + \eta x_i x_j$

Any problems with this learning rule?

Hebbian Learning

- If neuron x_i repeatedly triggers x_j, the synaptic knob connecting both grows

- Mathematical model $w_{i,j} = w_{i,j} + \eta x_i x_j$

Any problems with this learning rule?

→ Fundamentally unstable
  - Every connection only gets stronger
  - Nothing gets weaker (no reduction!)
  - No competition
  - Learning is unbounded
  - Various extensions (e.g. Sanger's rule)

Frank Rosenblatt's Perceptron

Frank Rosenblatt

Psychologist, Logician

Inventor of the solution to everything (i.e. the Perceptron)

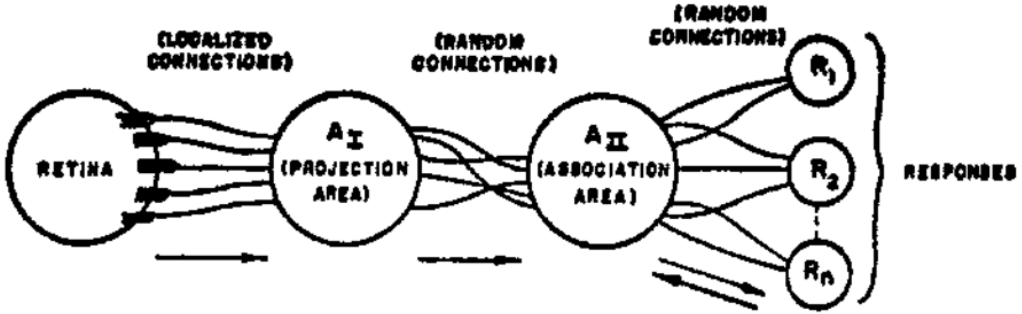
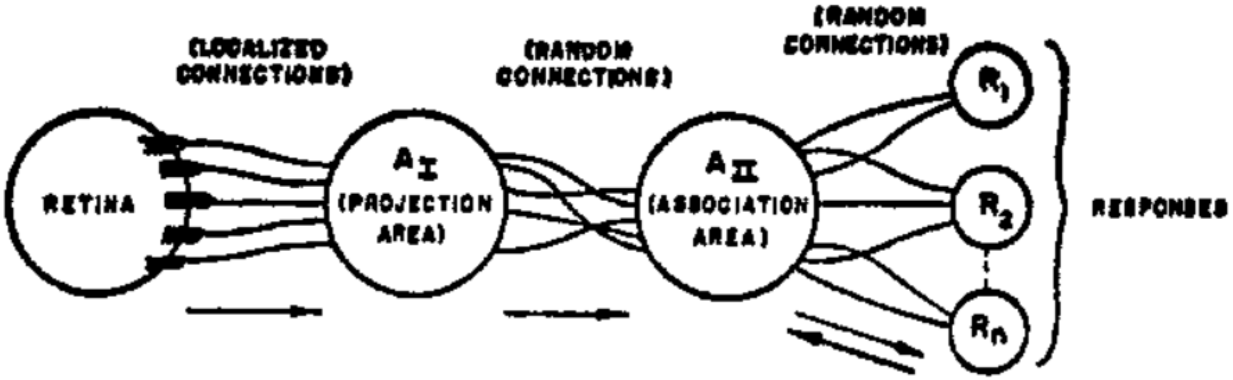


FIG. 1. Organization of a perceptron.

[Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.]

Frank Rosenblatt's Perceptron



- Groups of retinal sensors combine onto cells in association area A1
- Groups of A1 cells combine onto association cells A2
- Signals from A2 cells combine into response cells R

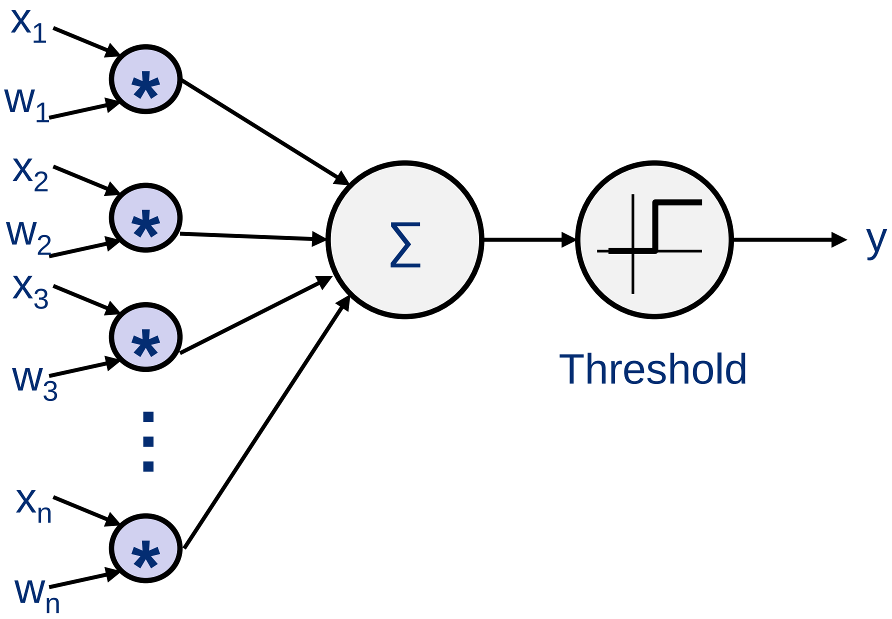
→ Connections may be excitatory or inhibitory

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

Rosenblatt's Perceptron: Mathematical Model

- Inputs combine linearly
- For now: Threshold logic

$$y = \begin{cases} 1, & \text{if } \sum_i w_i x_i - T > 0 \\ 0, & \text{else} \end{cases}$$



Rosenblatt's Perceptron: Mathematical Model

- Assumption: Can model any Boolean circuit!

- Given 11 inputs:
  - Build Boolean circuit that fires if majority of inputs are 1
  - Only using and, or gates, such circuit is exponential in size (consider all possible combinations)
  - With perceptron: Just does it!

Consequently, Rosenblatt was heavily funded!

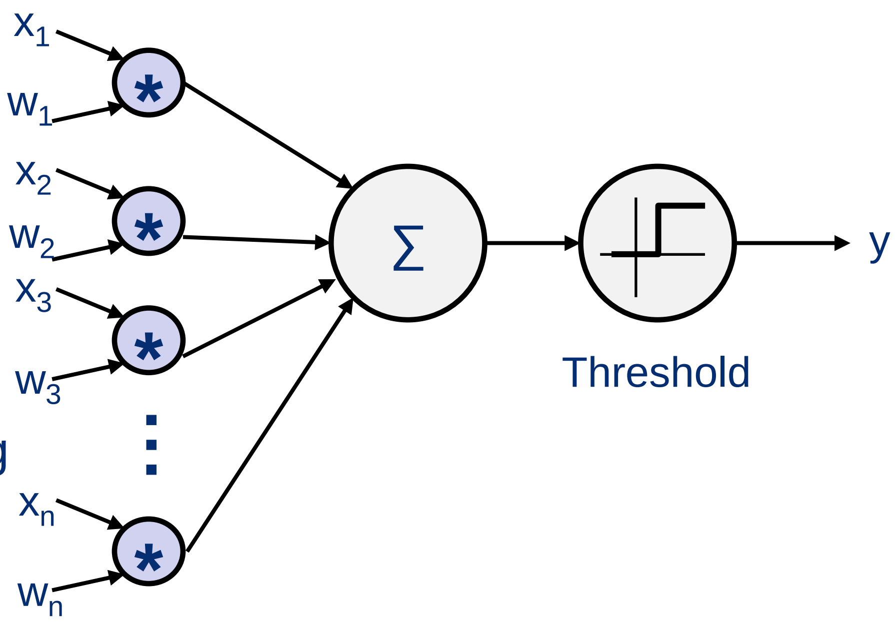
Navy made a lot of fuzz:

→ "… *the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence*"

Rosenblatt's Perceptron: Learning Algorithm

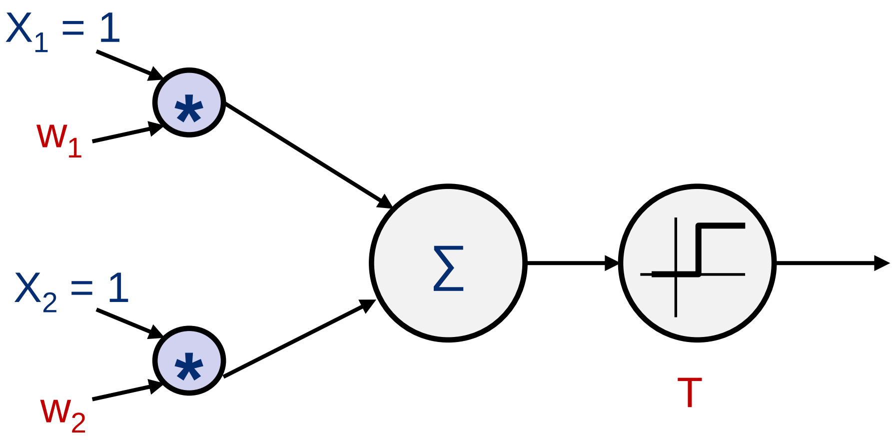
- Update rule
  $$w = w + \lambda(d(x) - y(x))$$
- Desired output: $d(x)$
- Actual output: $y(x)$

- Sequential learning
- Update weights when output is wrong
- Proved convergence if classes are linearly separable



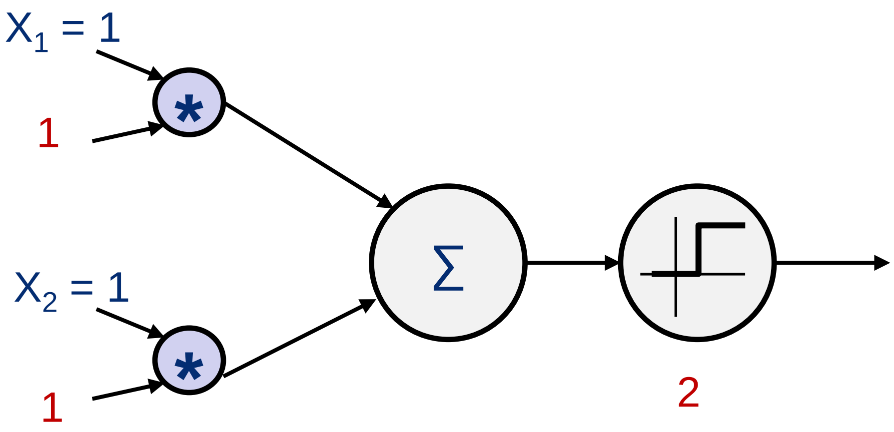
Perceptron: Boolean Functions

AND gate:



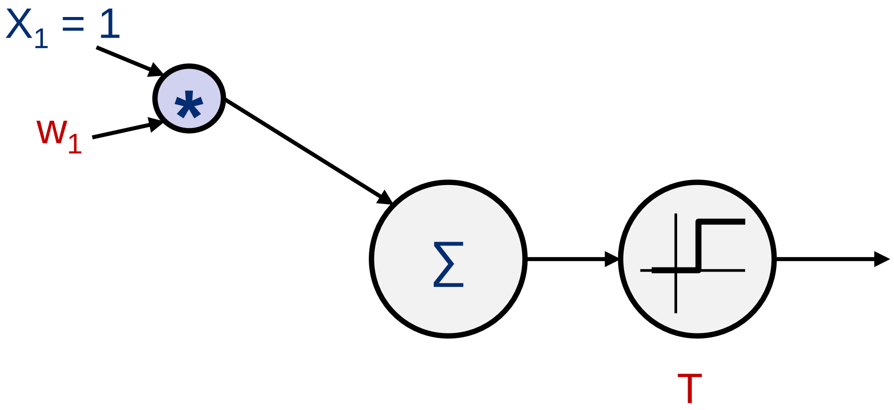
Perceptron: Boolean Functions

AND gate:



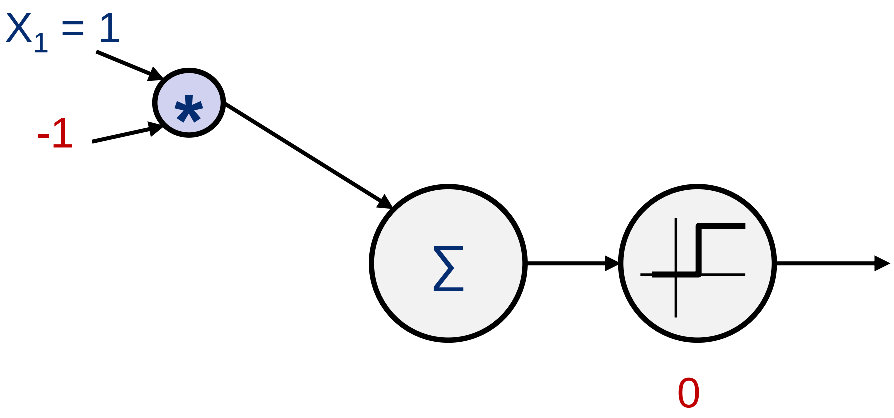
Perceptron: Boolean Functions

NOT gate:



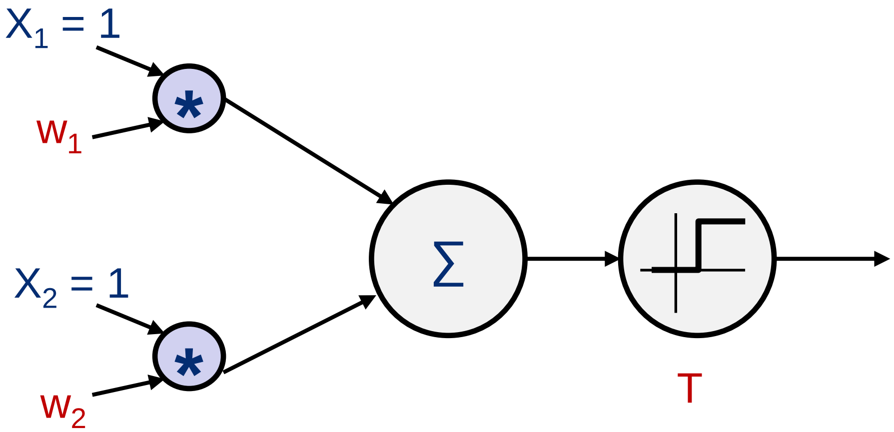
Perceptron: Boolean Functions

NOT gate:



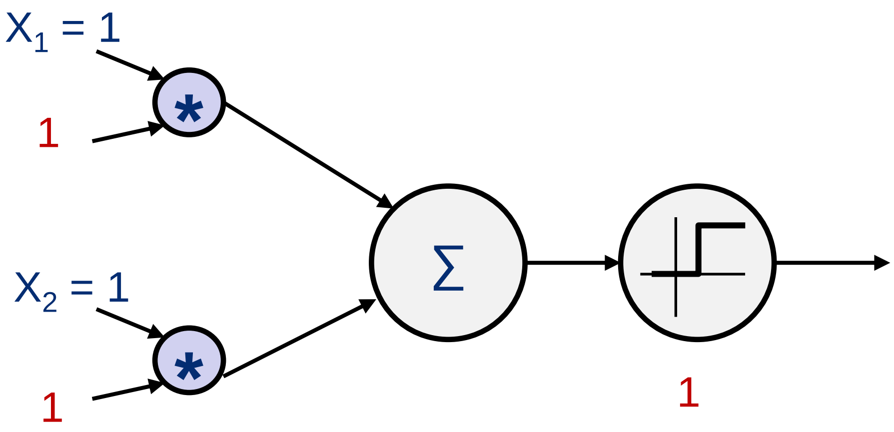
Perceptron: Boolean Functions

OR gate:



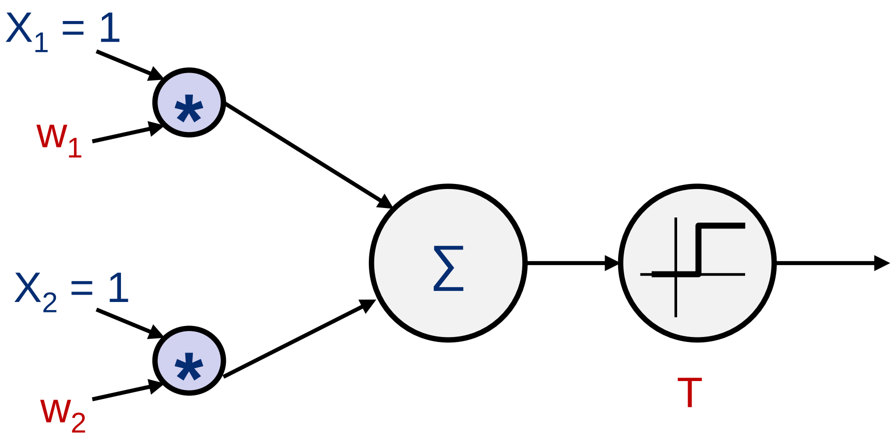
Perceptron: Boolean Functions

OR gate:



Perceptron: Boolean Functions

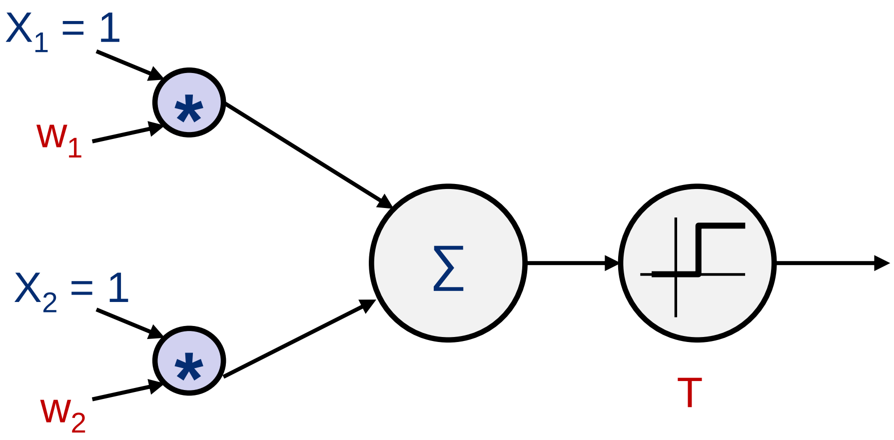
What about **XOR** gate:



Minsky, M., & Papert, S. (1969). Perceptron Expanded Edition. Only few lines in this book.

Perceptron: Boolean Functions

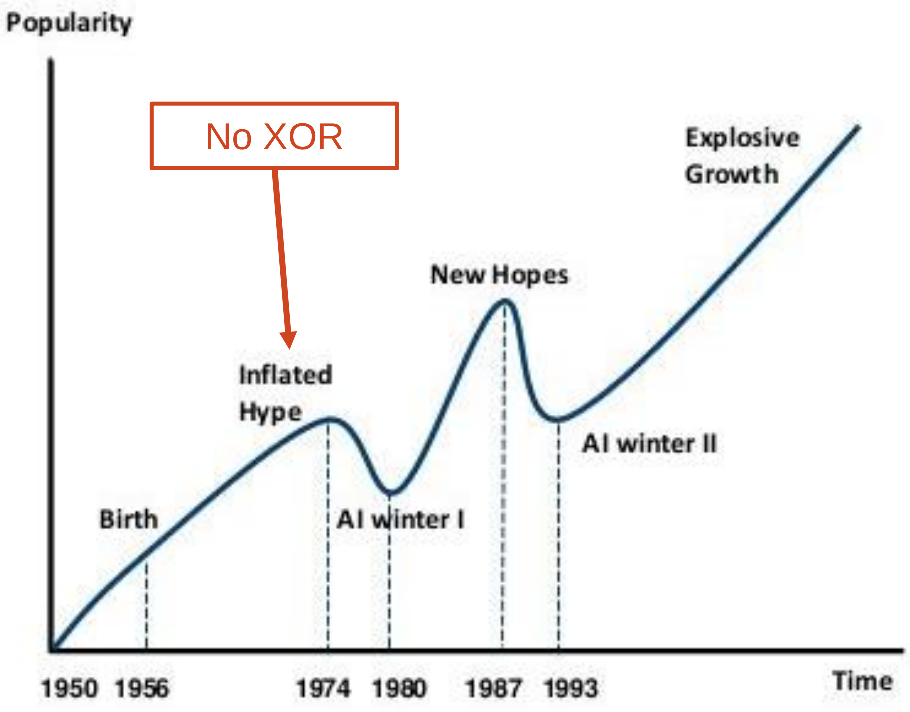
What about **XOR** gate:



→ No solution for XOR. Perceptrons are not universal Boolean circuit.

Minsky, M., & Papert, S. (1969). Perceptron Expanded Edition. Only few lines in this book.

A History of Being "The Next Big Thing"



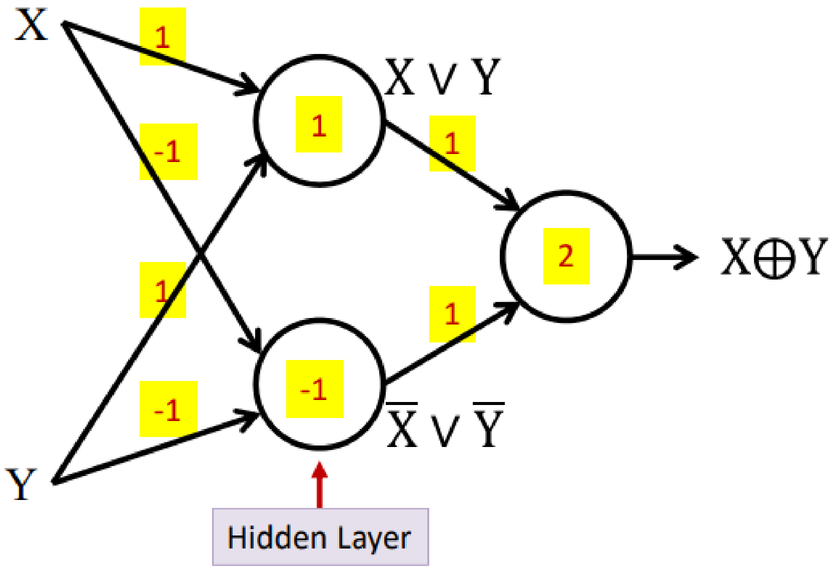
Timeline of AI Development

- **1950s-1960s**: First AI boom - the age of reasoning, prototype AI developed
- **1970s**: AI winter I
- **1980s-1990s**: Second AI boom: the age of Knowledge representation (appearance of expert systems capable of reproducing human decision-making)
- **1990s**: AI winter II
- **1997**: Deep Blue beats Gary Kasparov
- **2006**: University of Toronto develops Deep Learning
- **2011**: IBM's Watson won Jeopardy
- **2016**: Go software based on Deep Learning beats world's champions

Taken from Links International.

A Single Perceptron is Not Enough

Individual perceptrons are weak → Networked elements required!

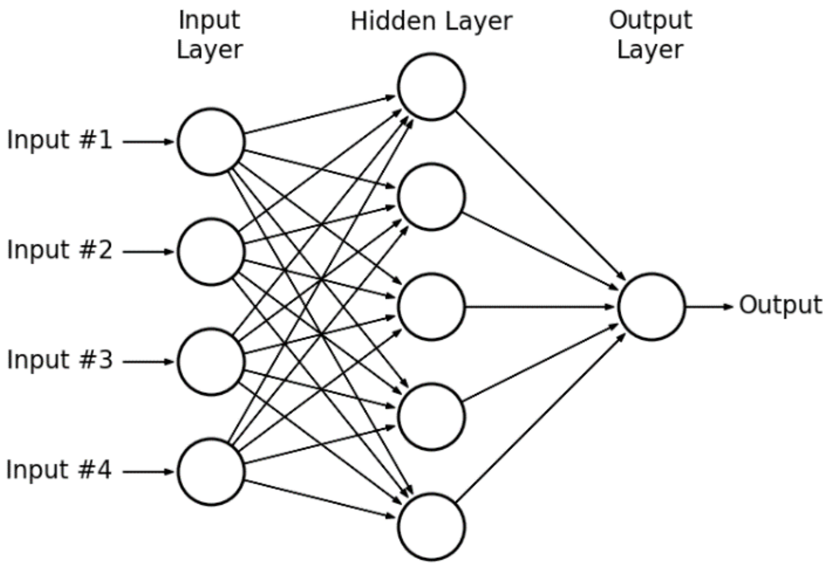


(X or Y) and (Not X or Not Y) → XOR

Minsky, M., & Papert, S. (1969). Perceptron Expanded Edition.

Multi-layer Perceptron

Individual perceptrons are weak → Networked elements required!



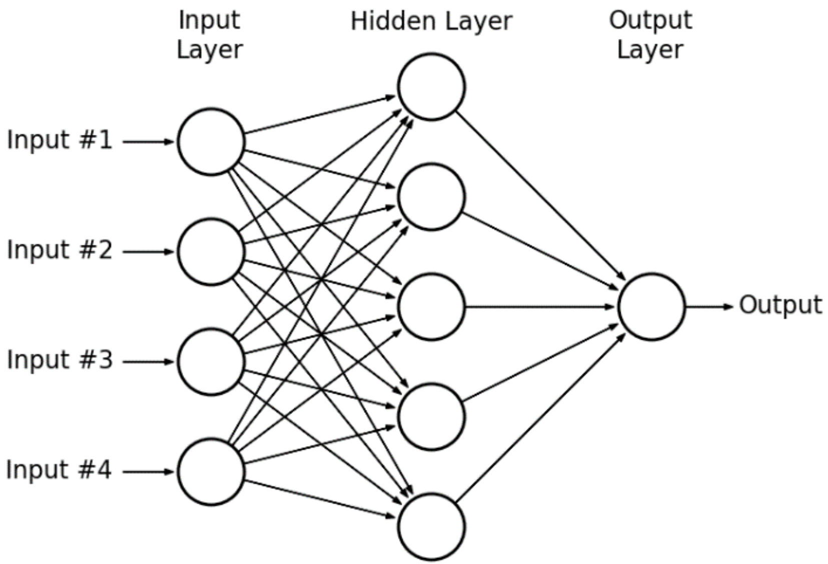
Can compose arbitrarily complicated Boolean functions!

Recap

- McCullough and Pitts model
  - Excitatory and inhibitory synapses
  - No learning rule
- Hebbian Learning
  - Neurons that fire together, wire together!
  - Unstable!
- Rosenblatt’s perceptron
  - Convergent learning rule for linearly separable problems
  - Single perceptrons are limited (Minsky and Papert)
- Multi-layer perceptrons to model arbitrary Boolean functions

Multi-layer Perceptron

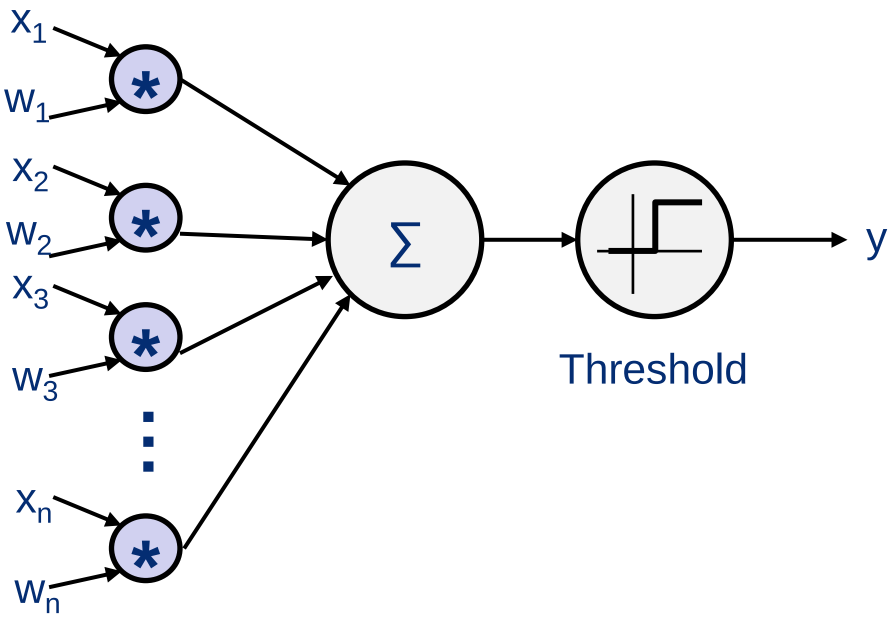
Multi-layer perceptrons can compose arbitrarily complex Boolean functions.



→ But our brain is not Boolean! Real inputs

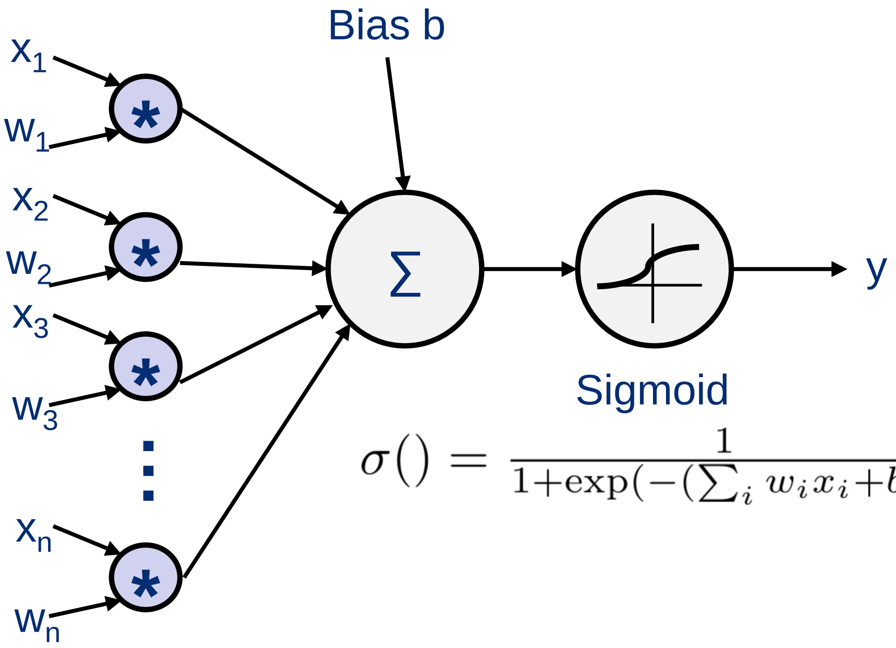
The Perceptron with Real-valued Inputs

- $\{x_i\}, \{w_i\}$ are real-valued
- Unit fires if weighted input exceeds threshold



The Perceptron with Real-valued Inputs and Outputs

- $\{x_i\}, \{w_i\}$ are real-valued
- Unit emits real-valued *activation*
- Choice of activation function!



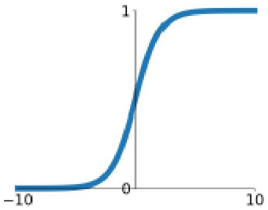
$$\sigma() = \frac{1}{1+\exp(-(\sum_i w_i x_i + b))}$$

The Perceptron with Real-valued Inputs and Outputs

We will discuss these in more detail later in the lecture.

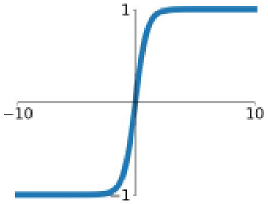
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



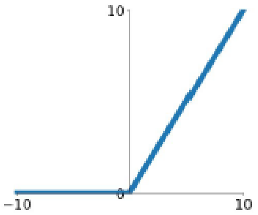
tanh

$\tanh(x)$



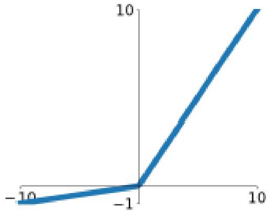
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

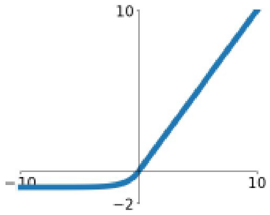


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

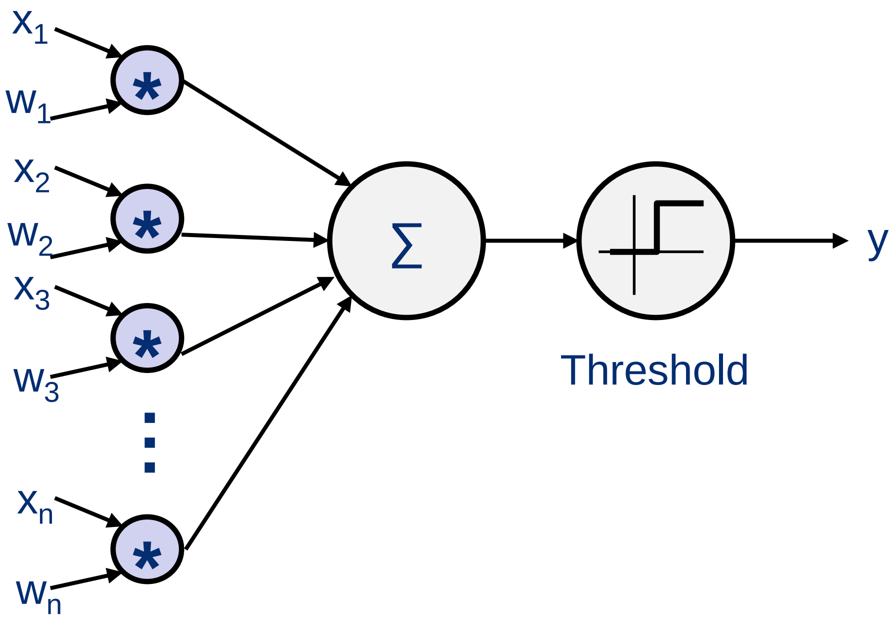
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

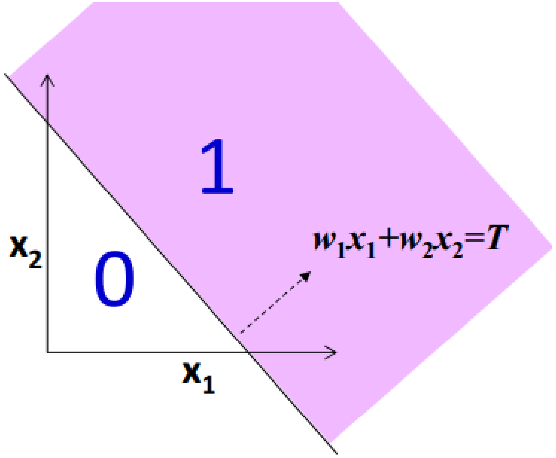


Boolean Functions with Real Perceptrons

- **For now**: Let's assume Boolean output for interpretation.
- Perceptrons operate on real-valued vectors and provide Boolean output
  → This is a linear classifier!



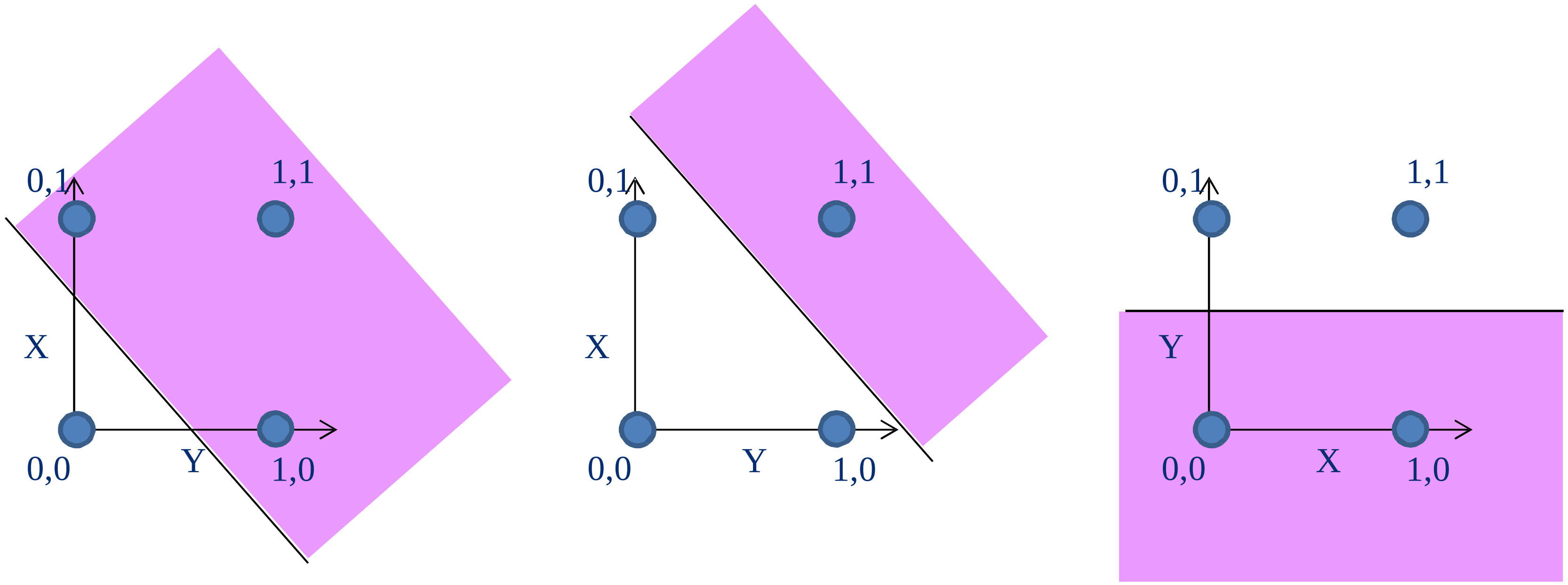
$$y = \begin{cases} 1, & \text{if } \sum_i w_i x_i - T > 0 \\ 0, & \text{else} \end{cases}$$



Content for following material adapted from CMU lecture.

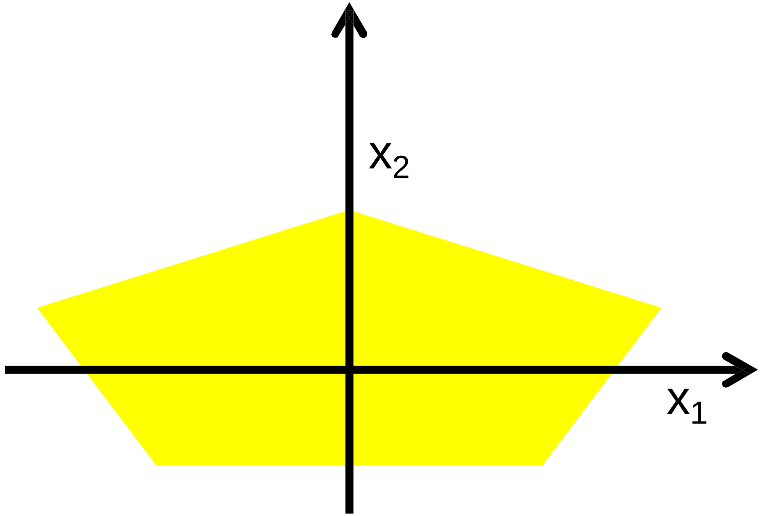
Boolean Functions with Real Perceptrons

Perceptrons are linear classifiers: Why no XOR?



Composing Complicated Decision Boundaries

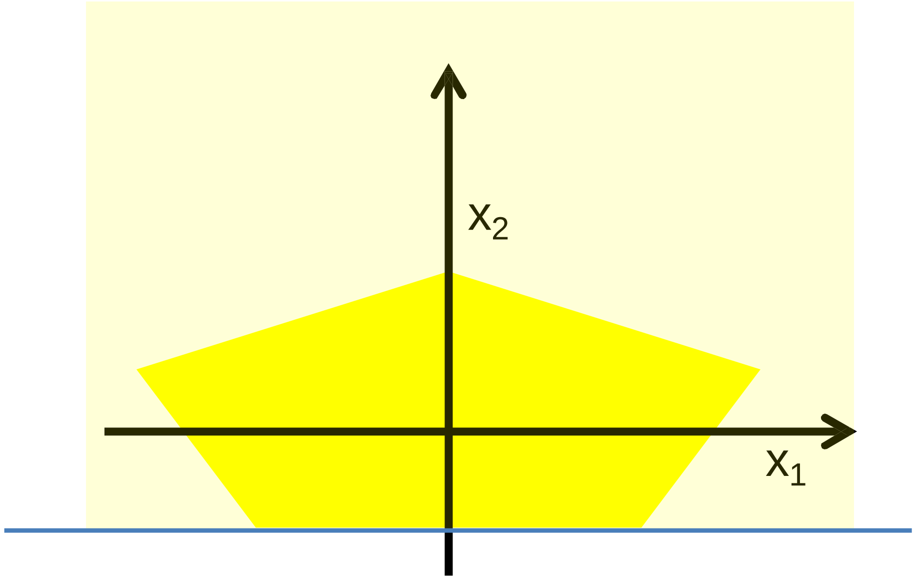
Task: Build a network of units with single output that fires in colored area

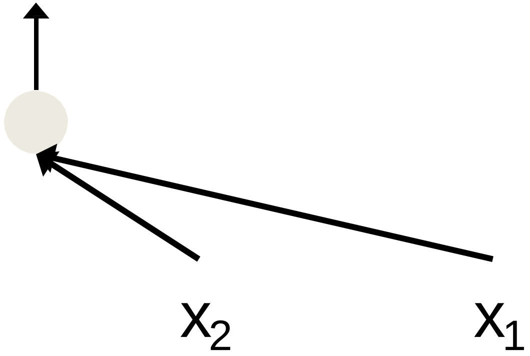


x_2 x_1

Composing Complicated Decision Boundaries

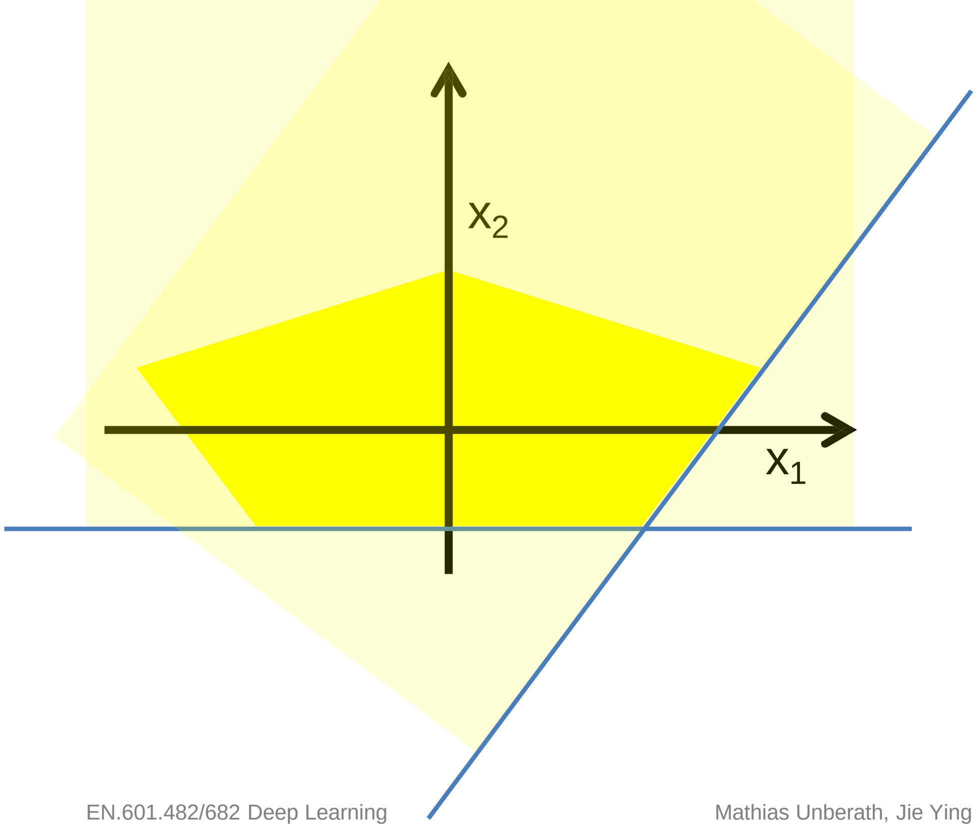
Task: Build a network of units with single output that fires in colored area

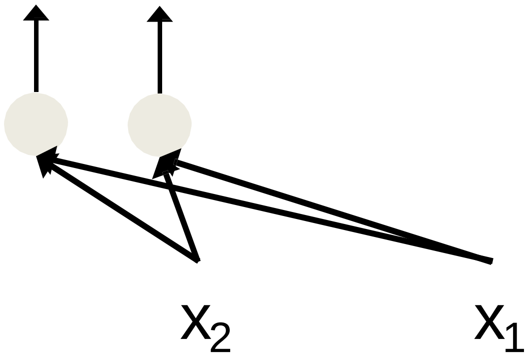




Composing Complicated Decision Boundaries

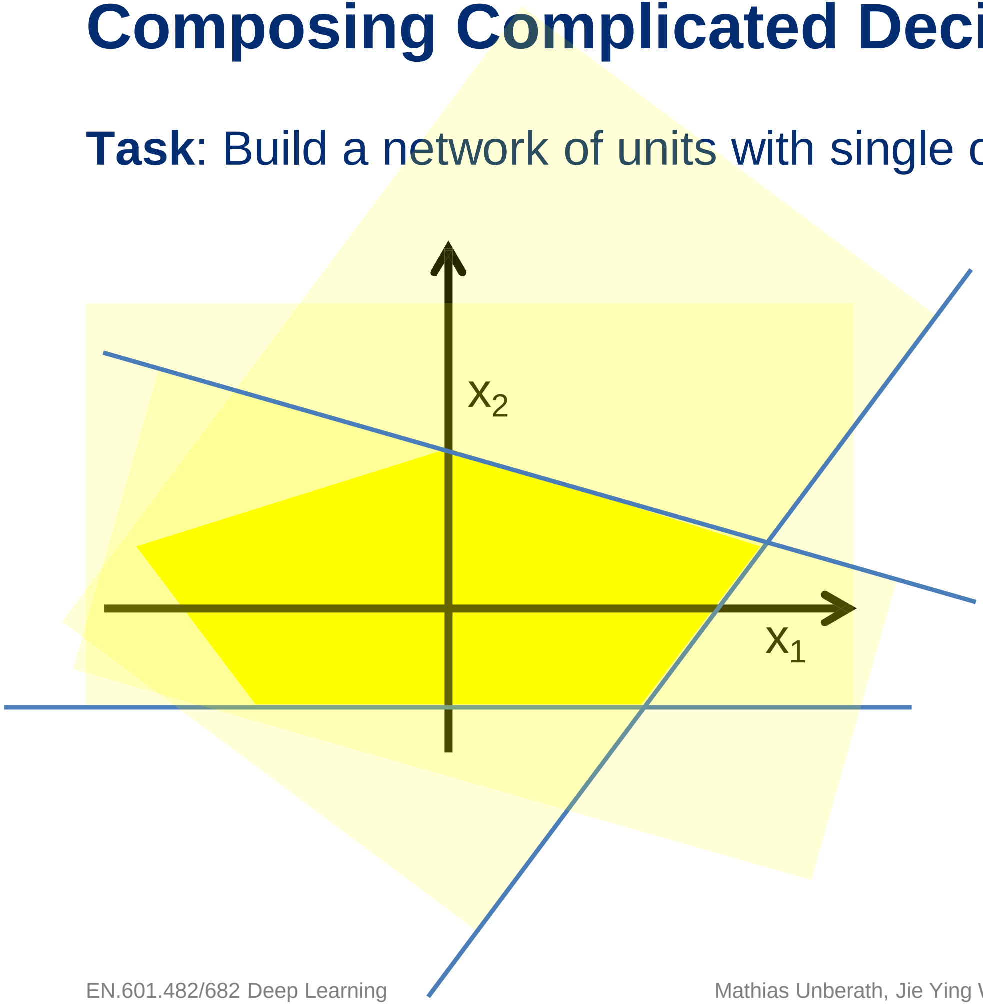
Task: Build a network of units with single output that fires in colored area

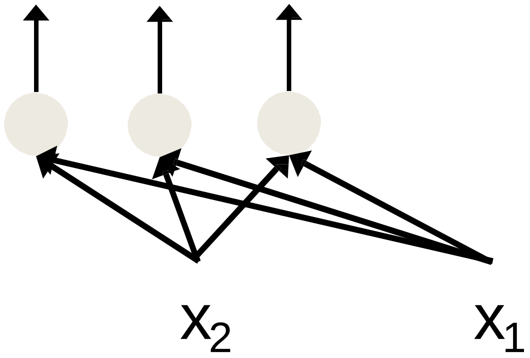




Composing Complicated Decision Boundaries

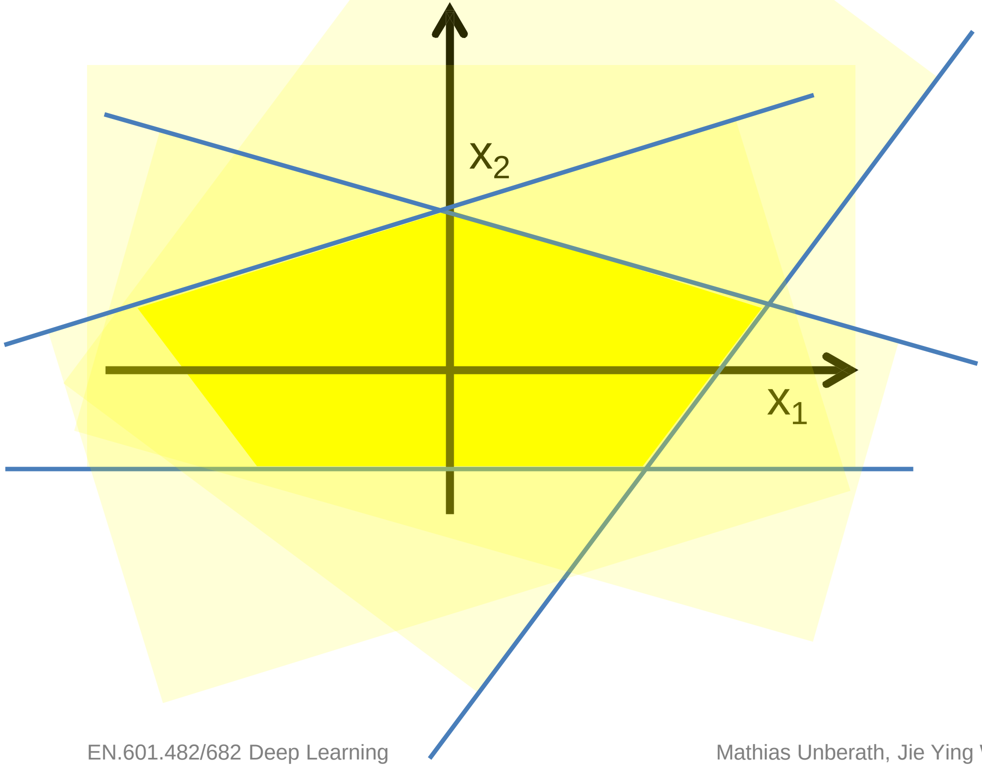
Task: Build a network of units with single output that fires in colored area

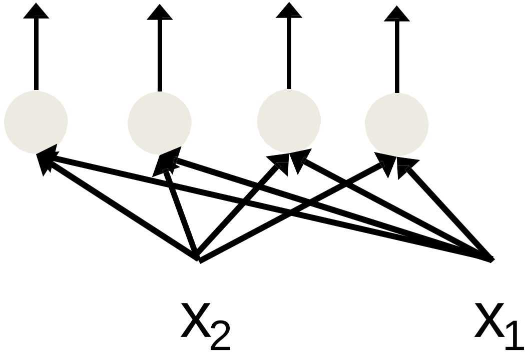




Composing Complicated Decision Boundaries

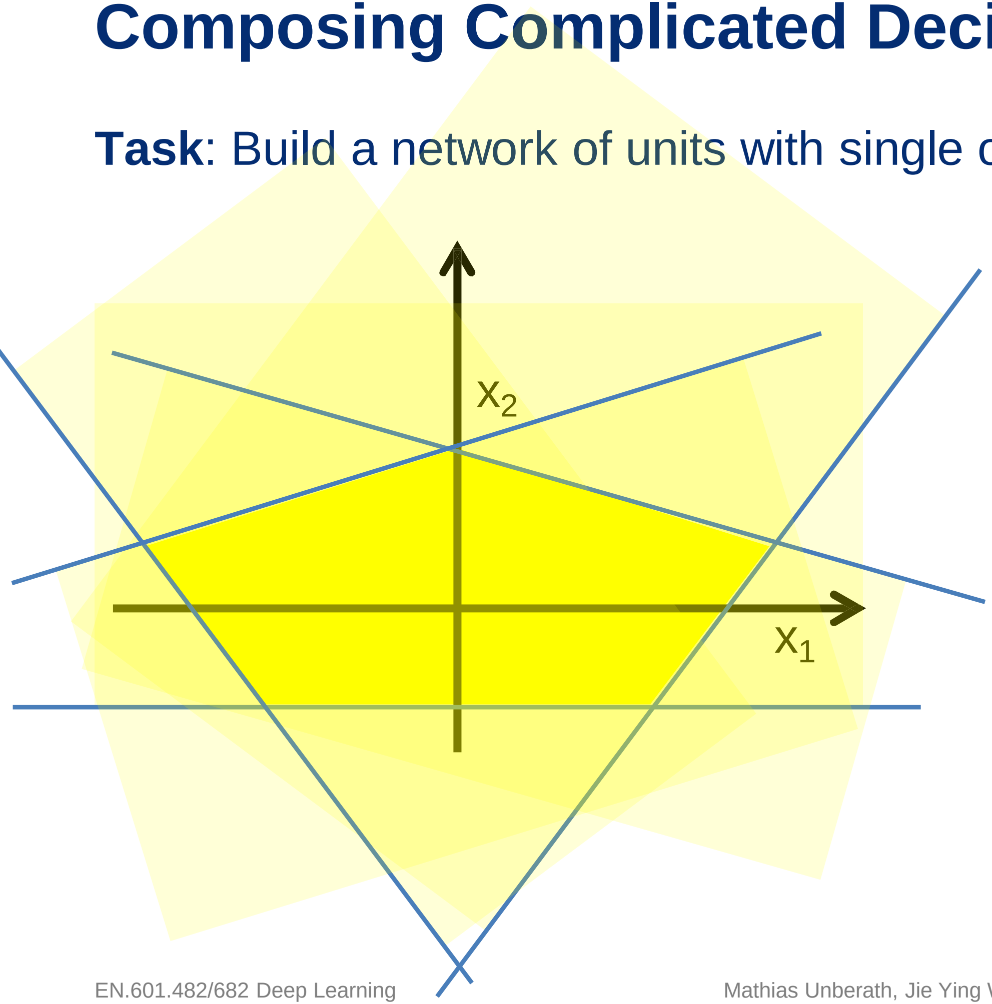
Task: Build a network of units with single output that fires in colored area

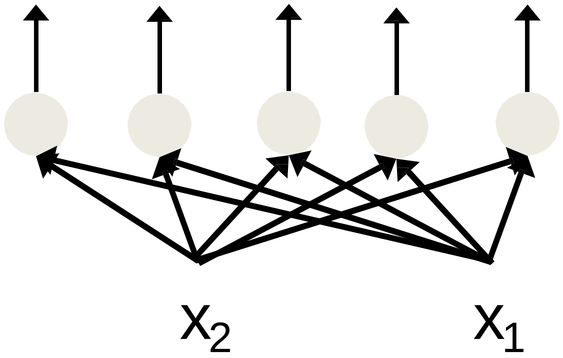




Composing Complicated Decision Boundaries

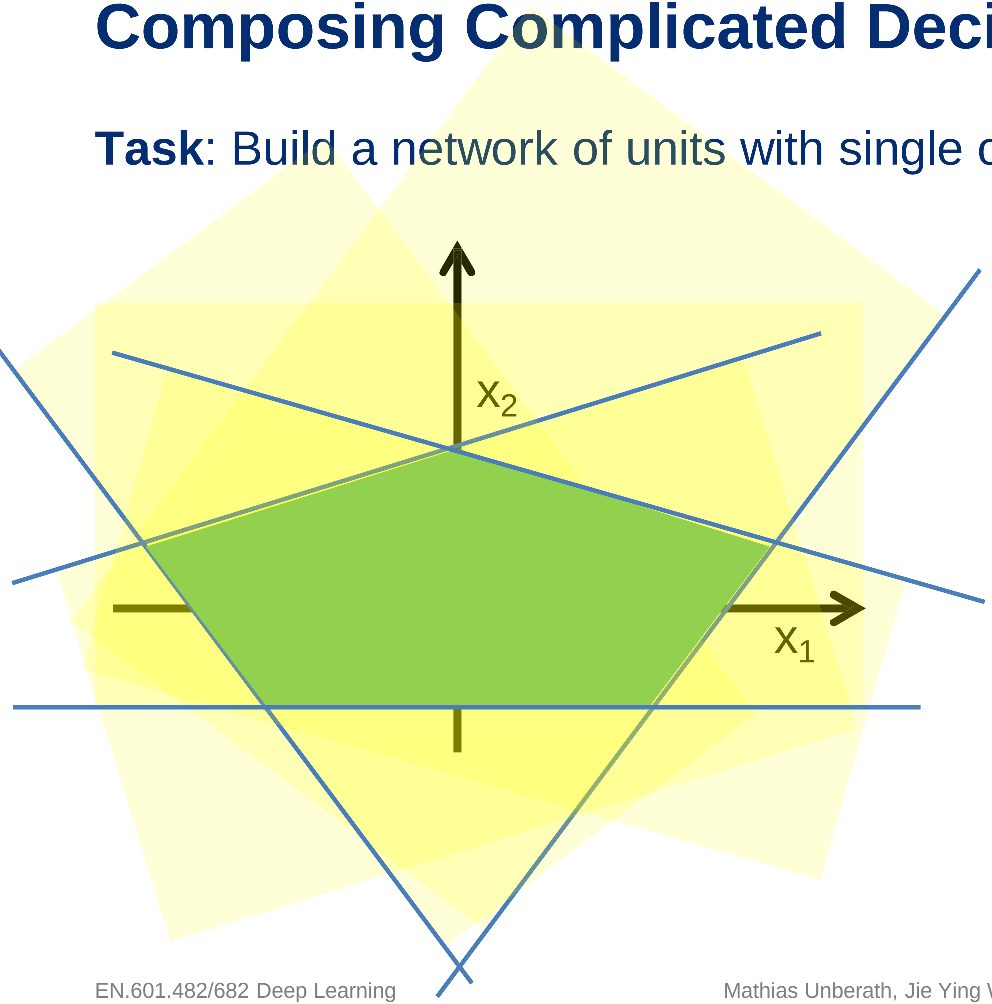
Task: Build a network of units with single output that fires in colored area

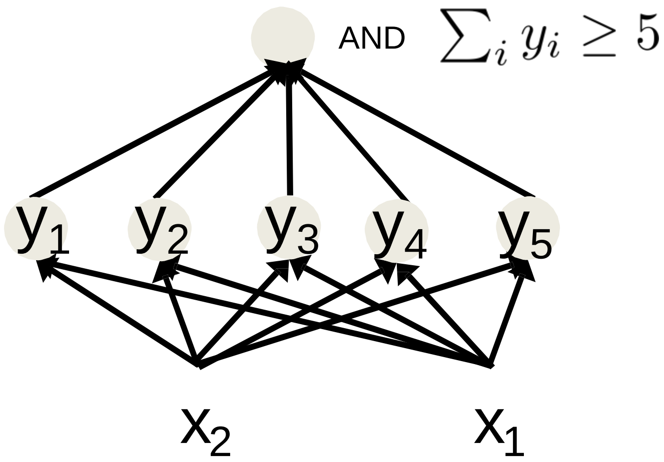




Composing Complicated Decision Boundaries

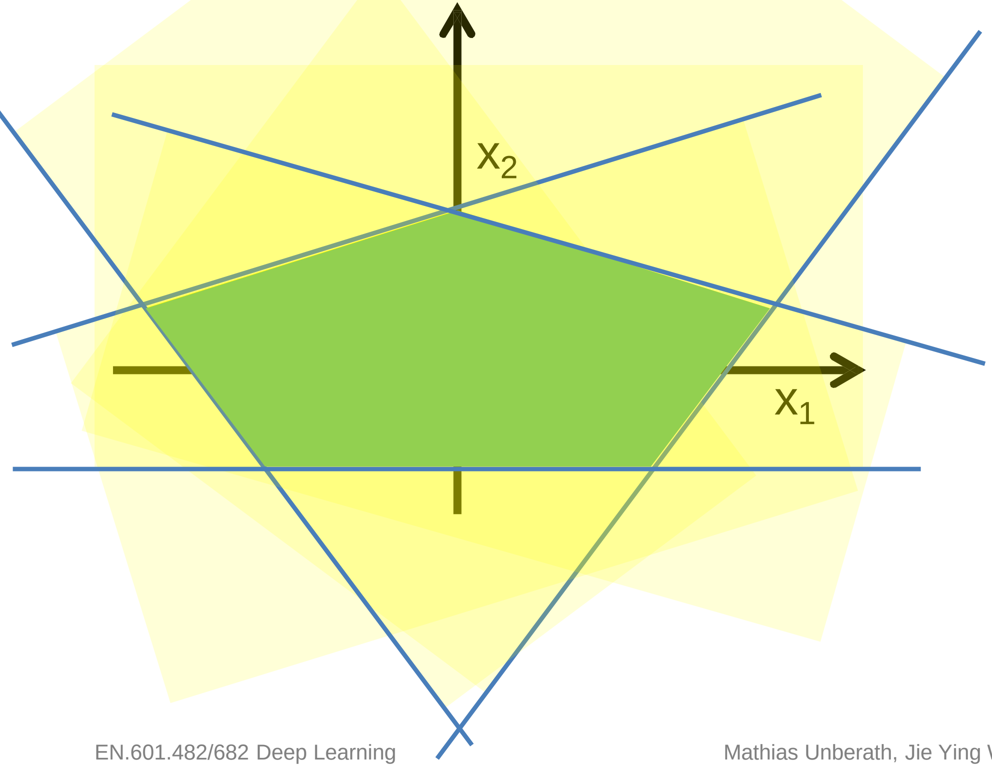
Task: Build a network of units with single output that fires in colored area



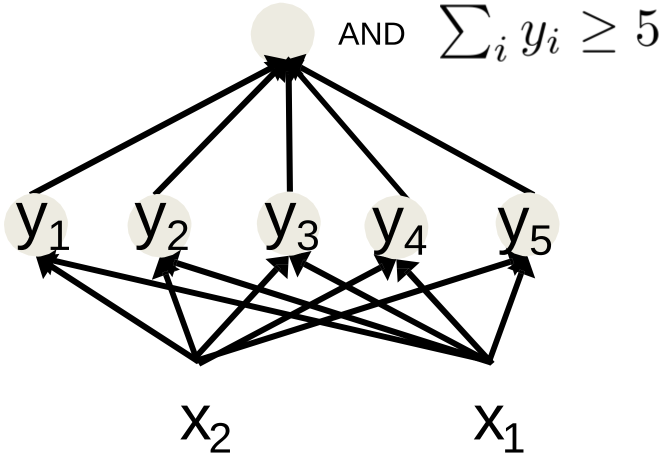


Composing Complicated Decision Boundaries

Task: Build a network of units with single output that fires in colored area

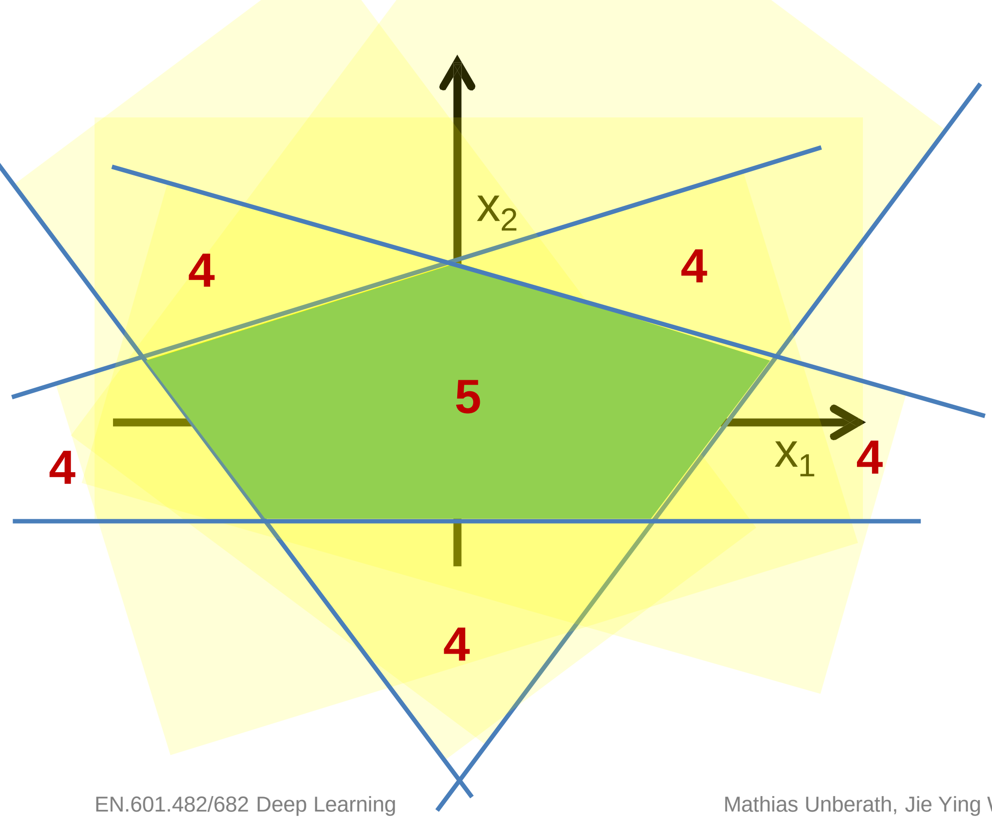


This shape is convex. Does this mean we can only construct convex shapes?

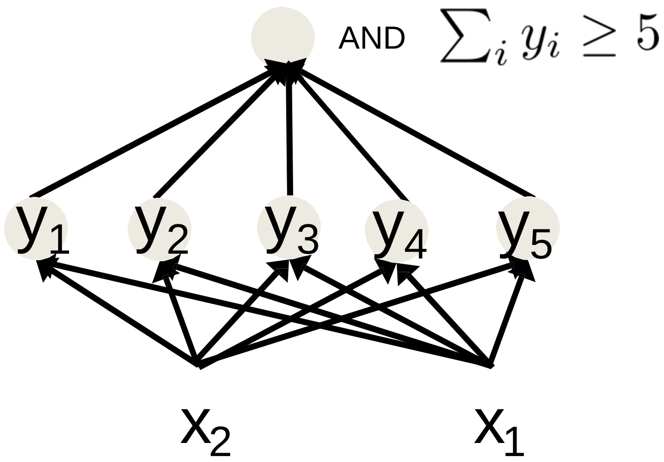


Composing Complicated Decision Boundaries

Task: Build a network of units with single output that fires in colored area

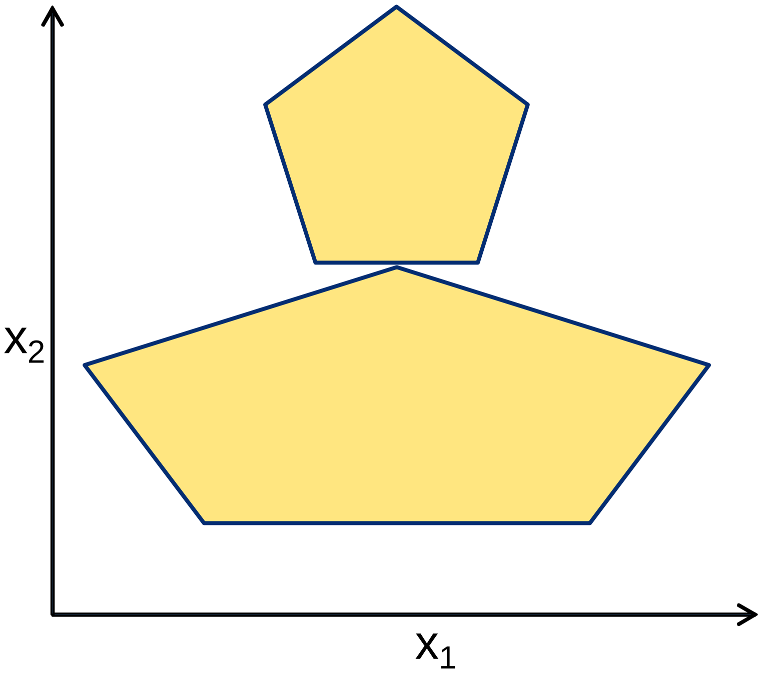


This shape is convex. Does this mean we can only construct convex shapes?



Even More Complex Decision Boundaries

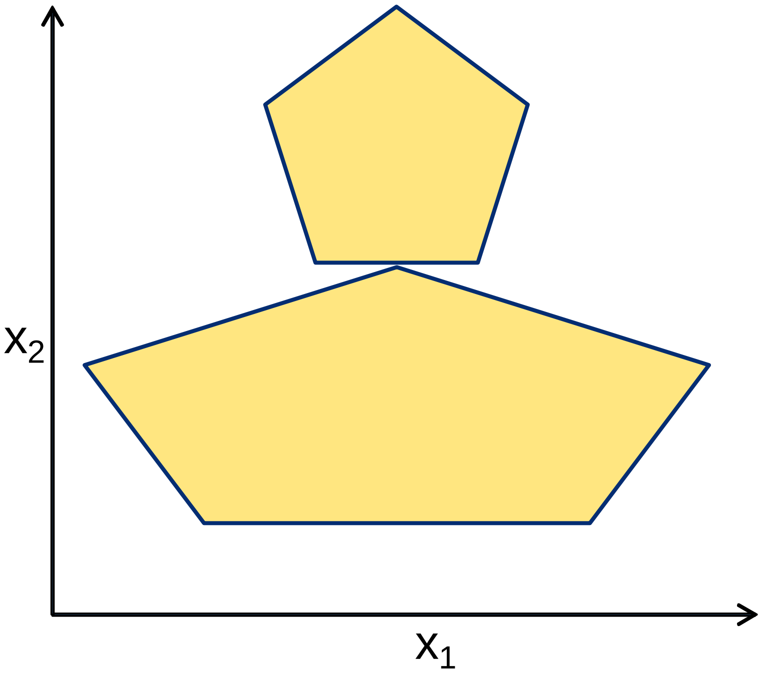
Task: Build network that fires in colored area

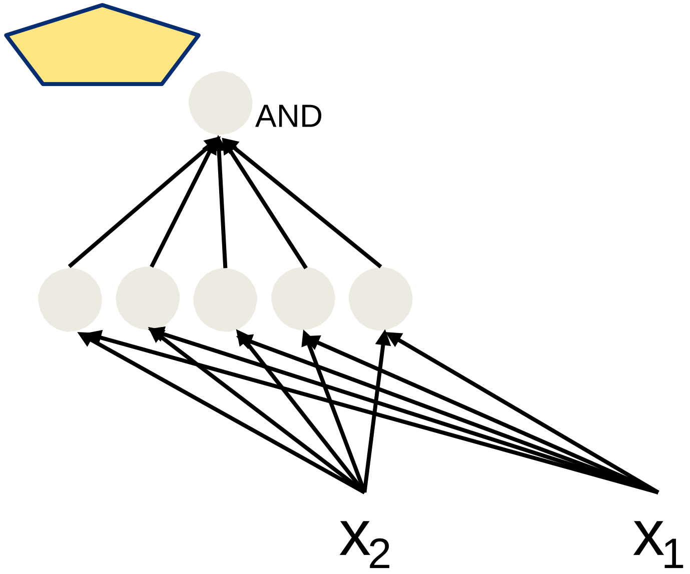


Q: How do we go about this?

Even More Complex Decision Boundaries

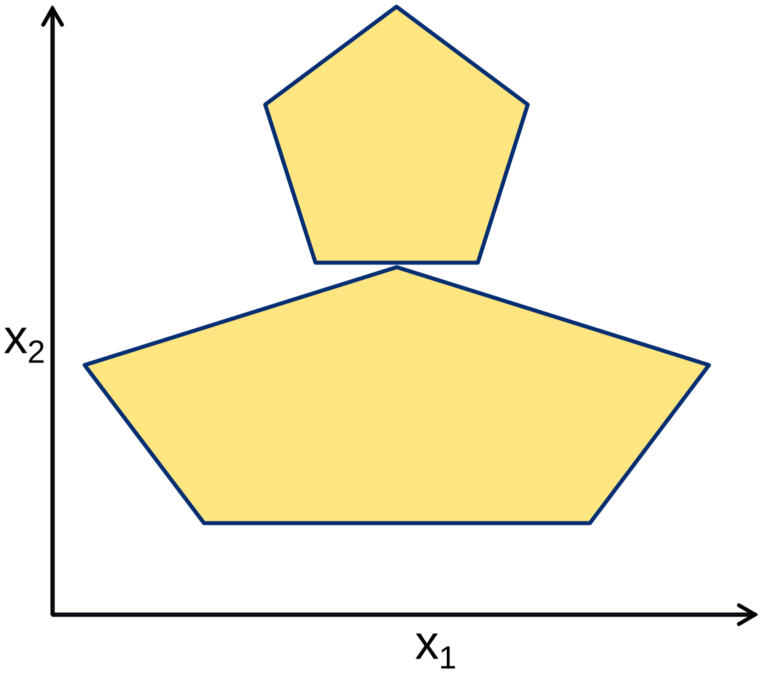
Task: Build network that fires in colored area

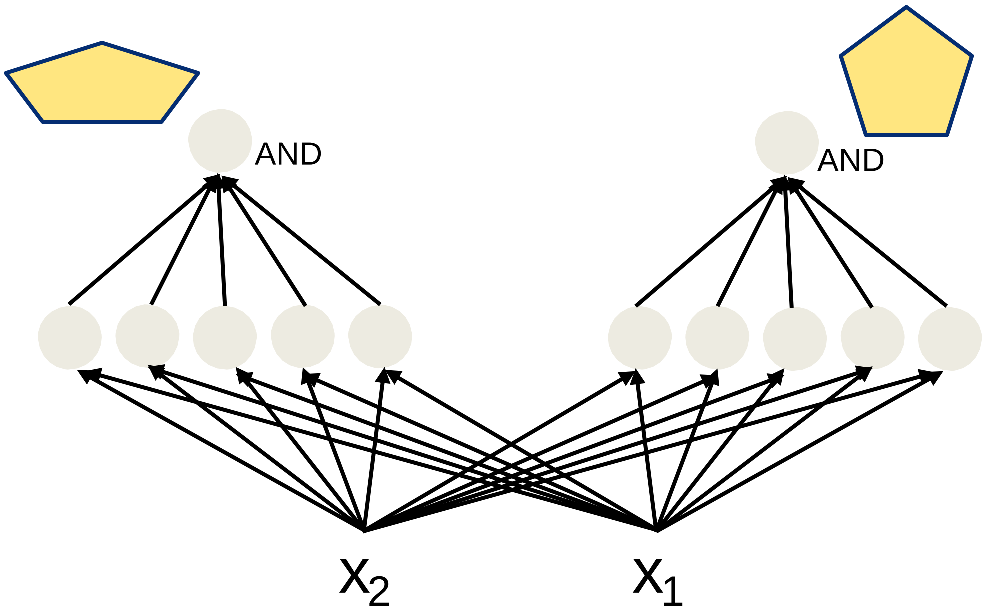




Even More Complex Decision Boundaries

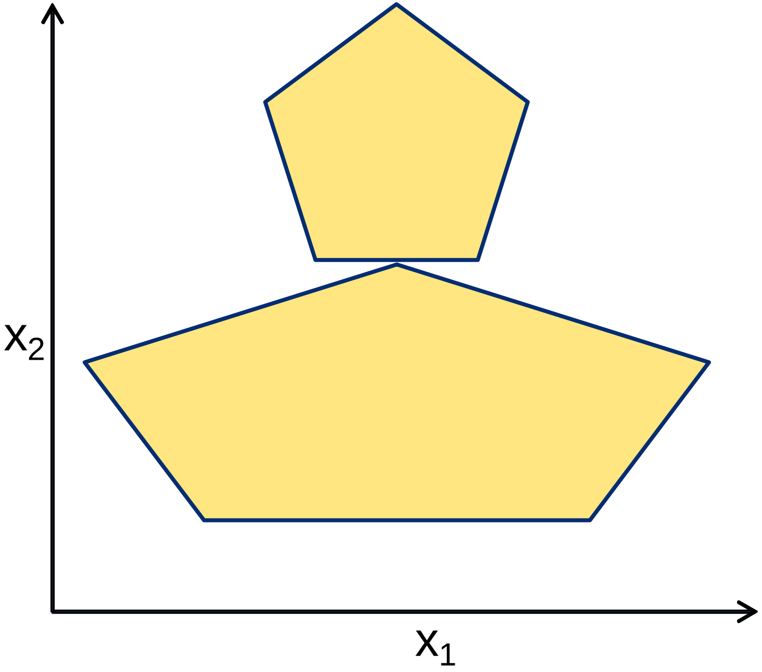
Task: Build network that fires in colored area

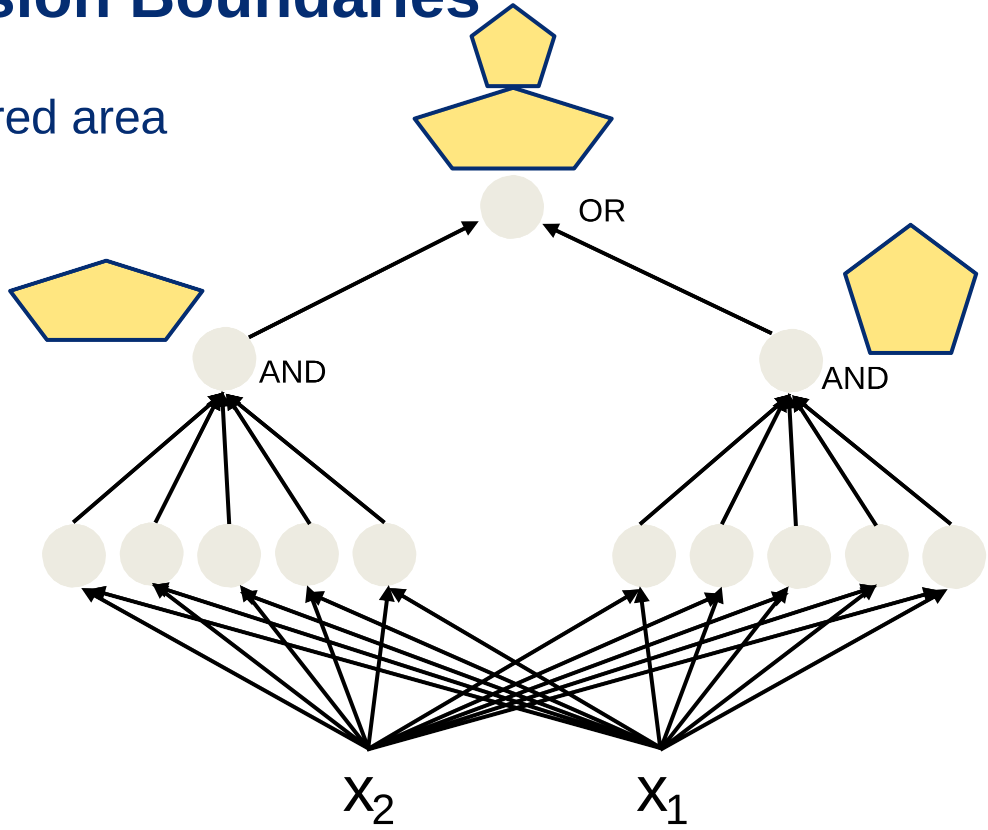




Even More Complex Decision Boundaries

Task: Build network that fires in colored area





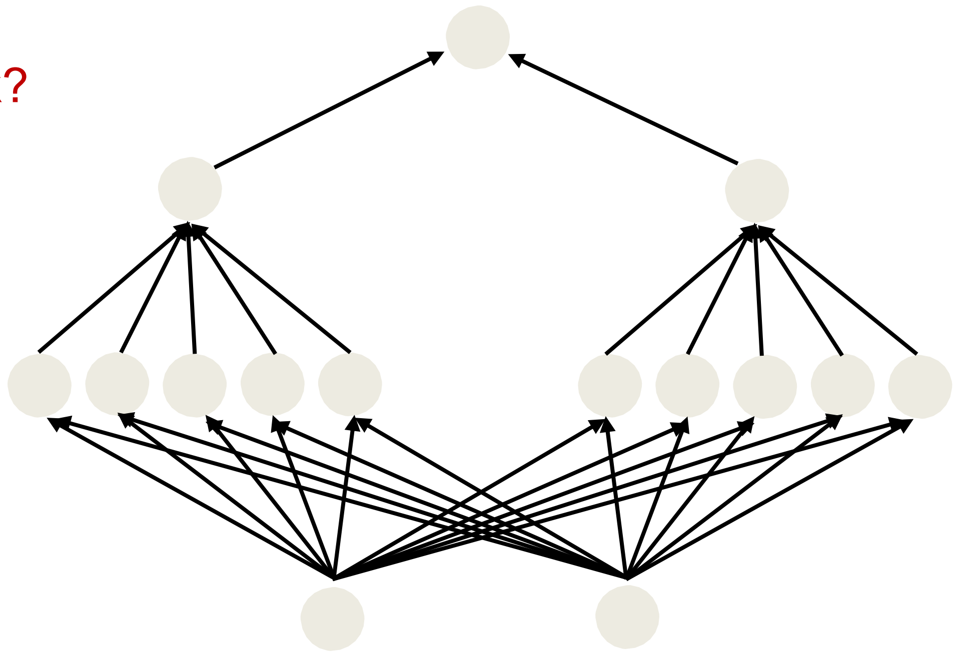
Recap

- Rosenblatt’s perceptron
  - Convergent learning rule for linearly separable problems
  - Single perceptrons are limited (Minsky and Papert)
- Multi-layer perceptrons can model arbitrary Boolean functions
- Multi-layer neural networks can model arbitrarily complex decision boundaries

The Question of Depth

- We commonly talk about **deep neural networks**

- Q: What is a "**deep**" neural network?

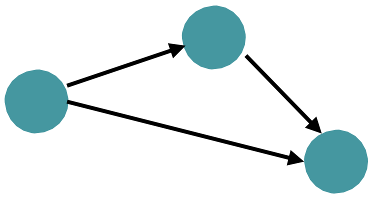


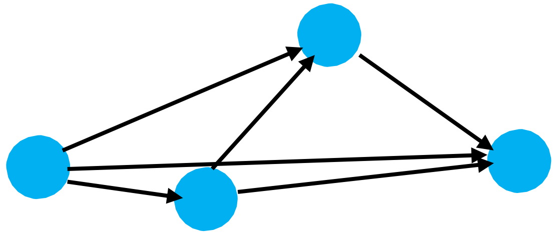
The Question of Depth

- We commonly talk about **deep neural networks**

- Q: What is a “**deep**” neural network?

- Directed network of computational elements
  - Inputs: Source nodes
  - Output: Sink nodes
  - Then, **depth**: Length of longest path from source to sink



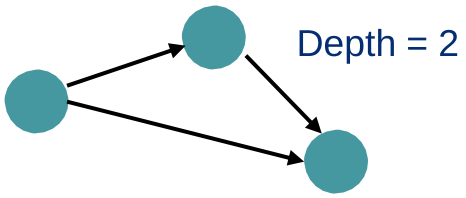


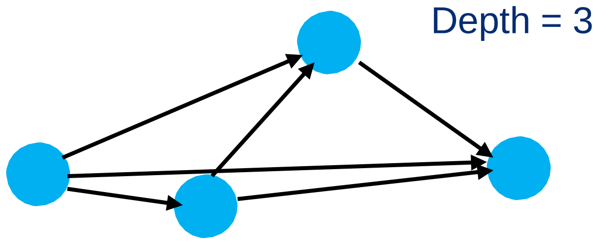
The Question of Depth

- We commonly talk about **deep neural networks**

- Q: What is a “**deep**” neural network?

- Directed network of computational elements
  - Inputs: Source nodes
  - Output: Sink nodes
  - Then, **depth**: Length of longest path from source to sink



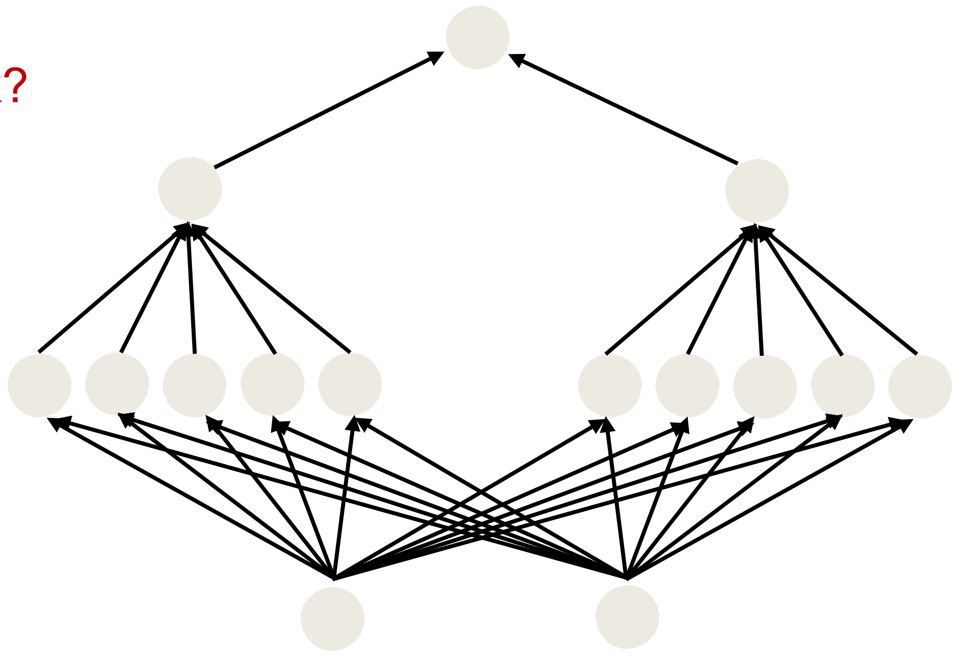


The Question of Depth

- We commonly talk about **deep neural networks**

- Q: What is a "**deep**" neural network?

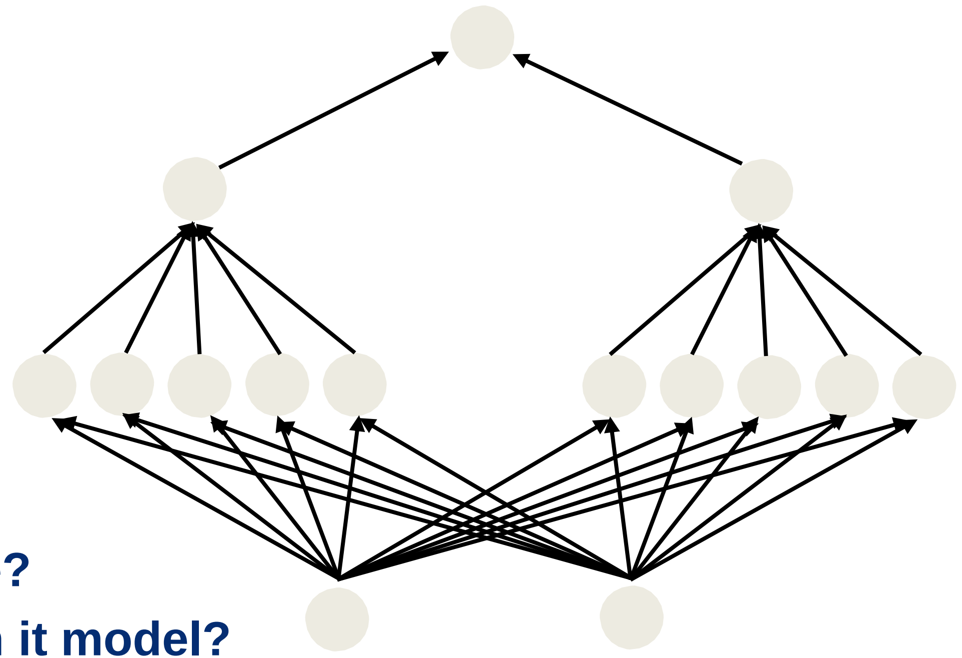
- Layered structure:
  "Deep" → Depth > 2



The Multi-layer Perceptron

- **Inputs** can be real or Boolean

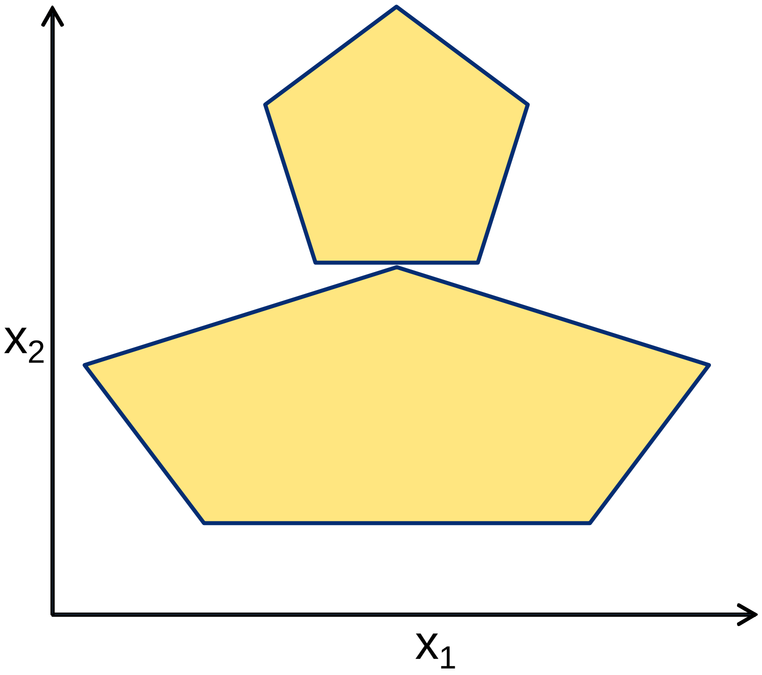
- **Outputs** can be real or Boolean

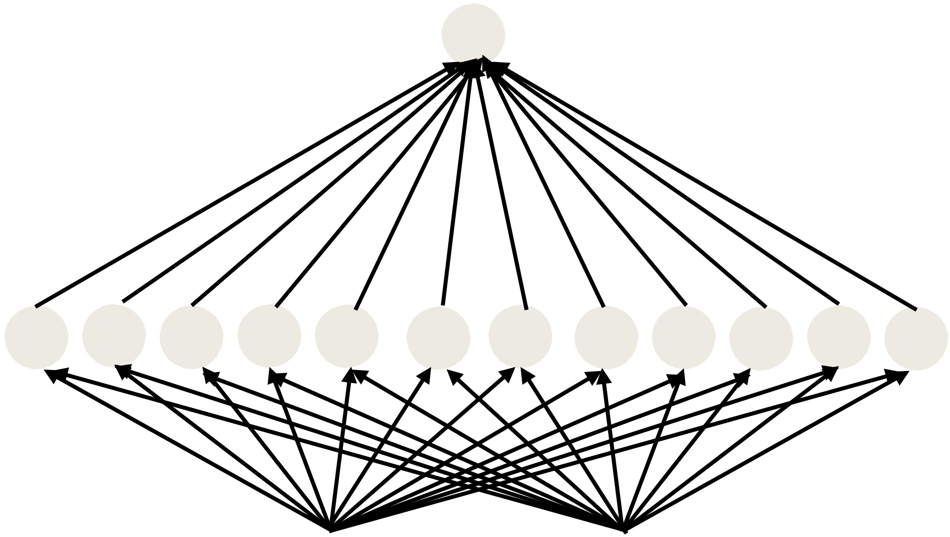


Q: What can such network compute?
What input/output relationships can it model?

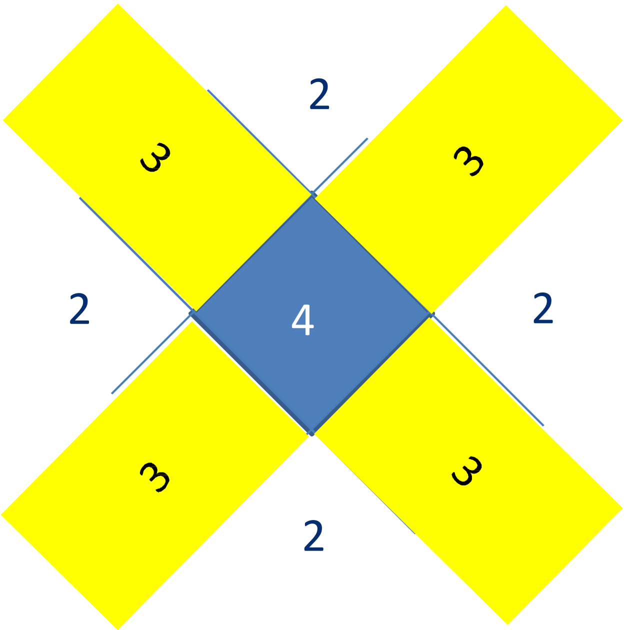
Even More Complex Decision Boundaries

Task: Build network that fires in colored area **with only one hidden layer**

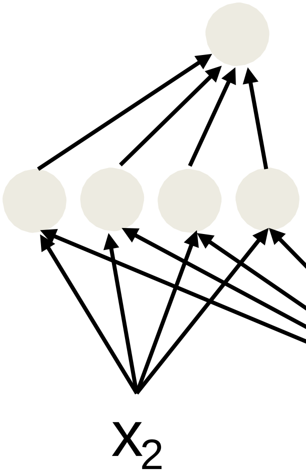




Square Decision Boundary

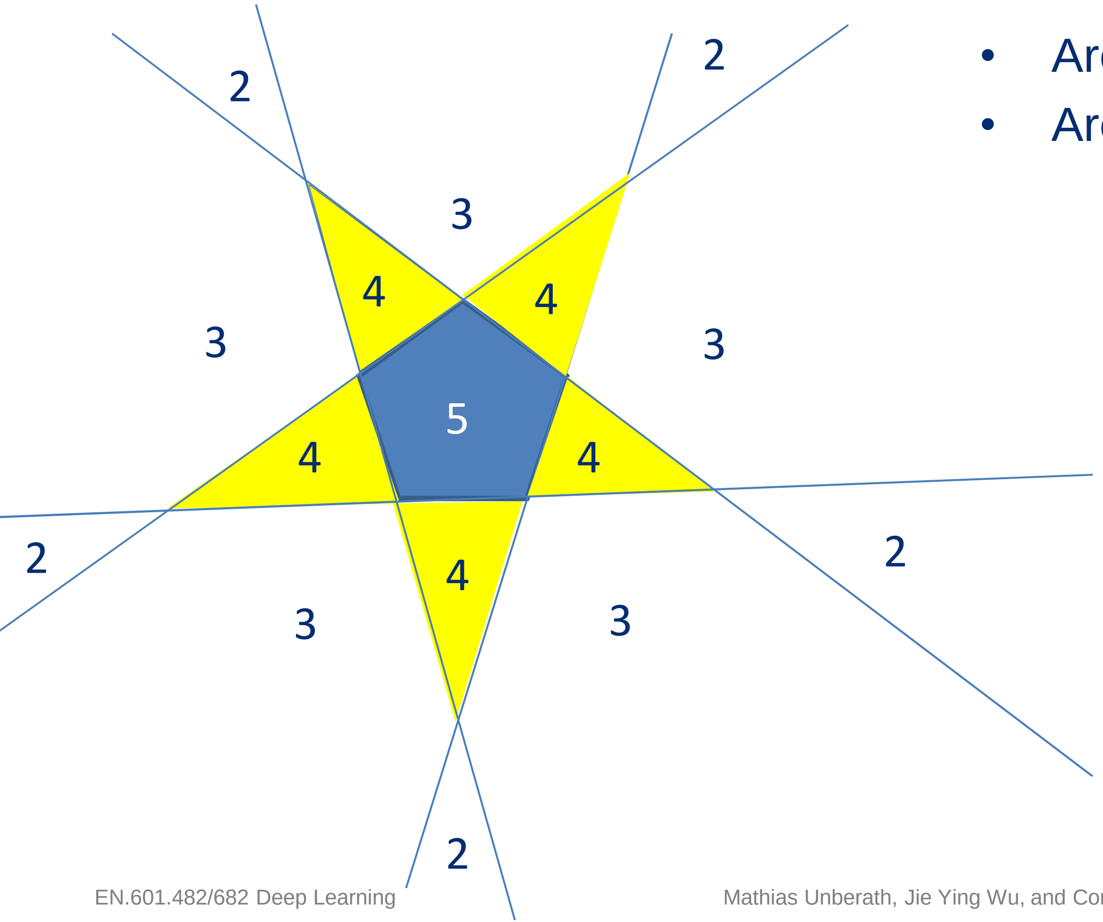


- Area of 4 is finite
- Area of 2, 3 is infinite

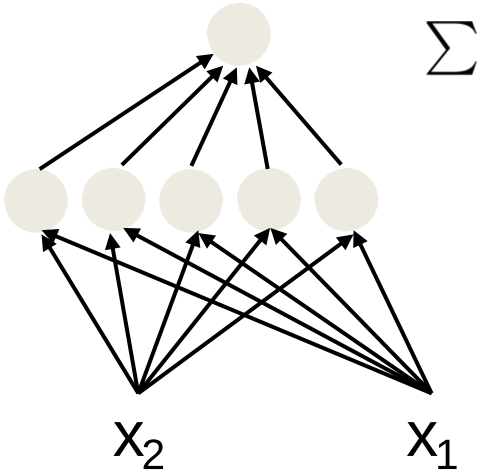


$$\sum_i y_i \geq 4$$

Composing a Pentagon

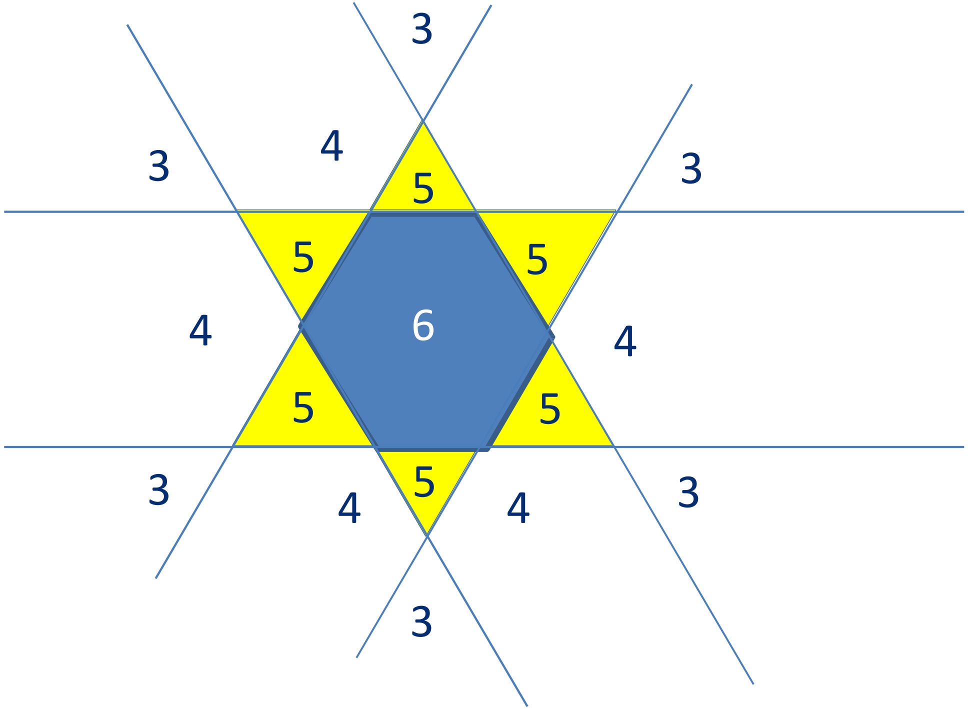


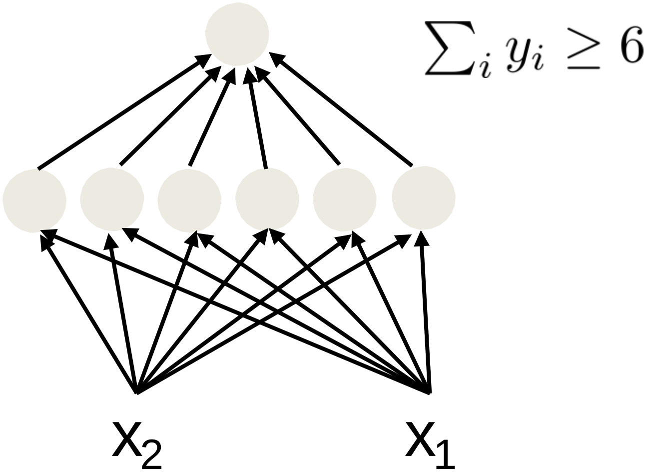
- Area of n-1 is finite
- Area of n-2 is infinite



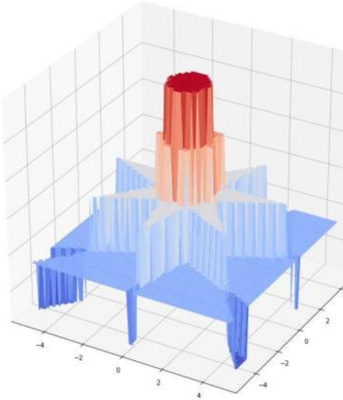
$$\sum_i y_i \geq 5$$

Composing a Hexagon

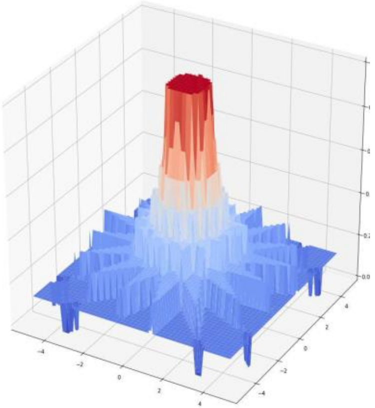




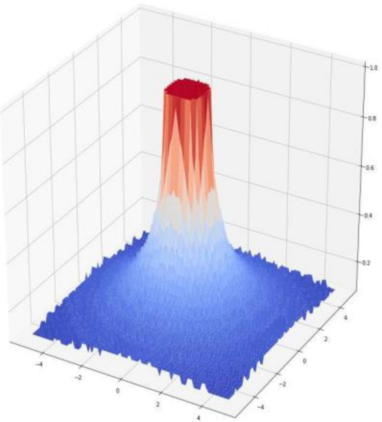
Pattern Emerges with N > 6



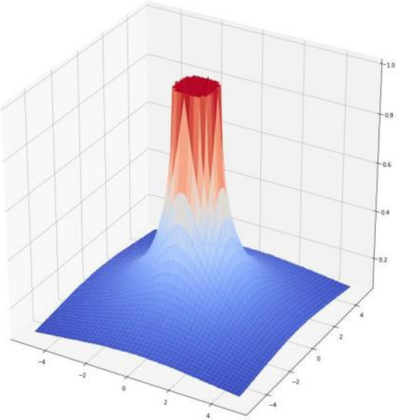
Heptagon



16 sides



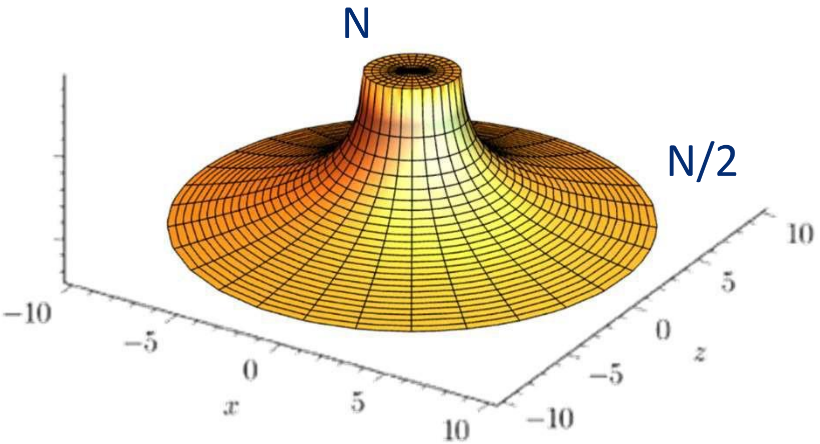
64 sides

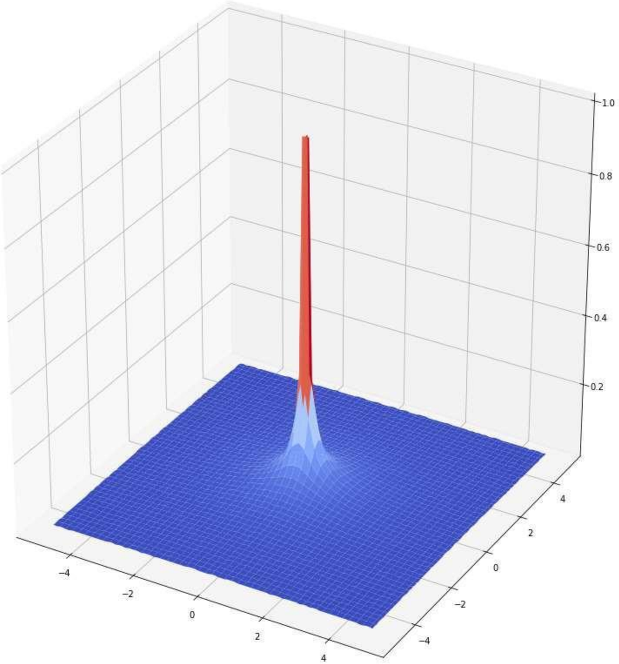


1000 sides

→ Increasing number of sides reduces the area outside polygon that have N/2 < sum < N

In the Limit



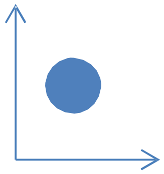


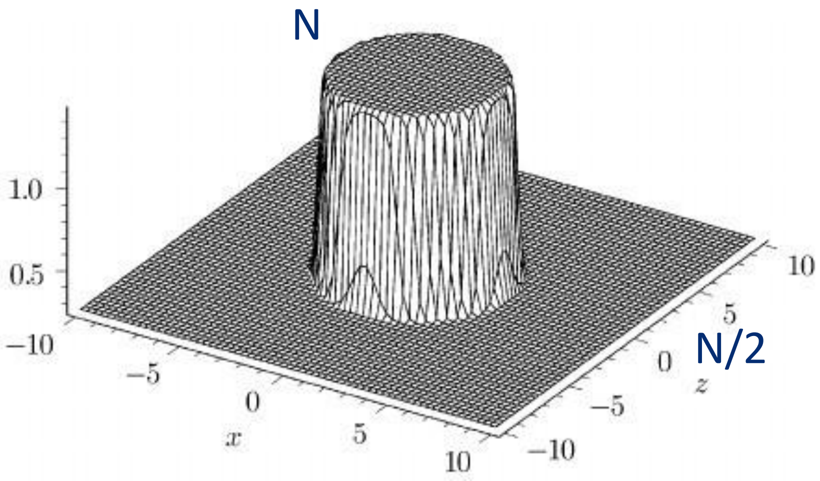
For small radius → near perfect cylinder!
N within the cylinder, N/2 outside

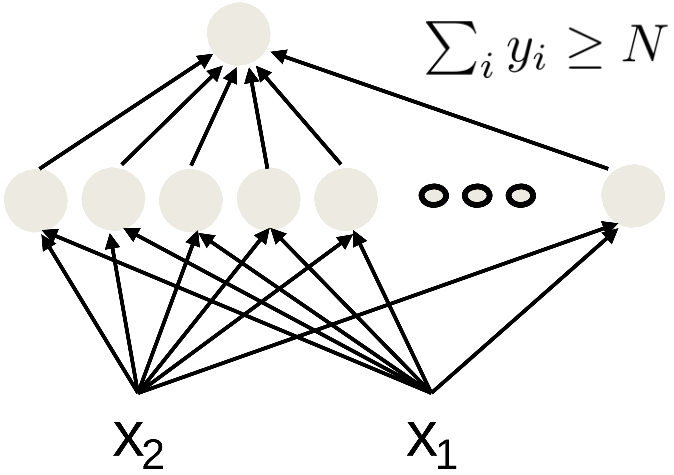
Composing a Circle

The circle network

- Very large number of neurons (→ infinitely many)
- Sum is N inside circle & N/2 almost everywhere outside
- Circle can be at **any** location!



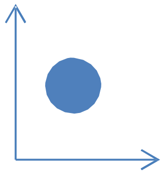


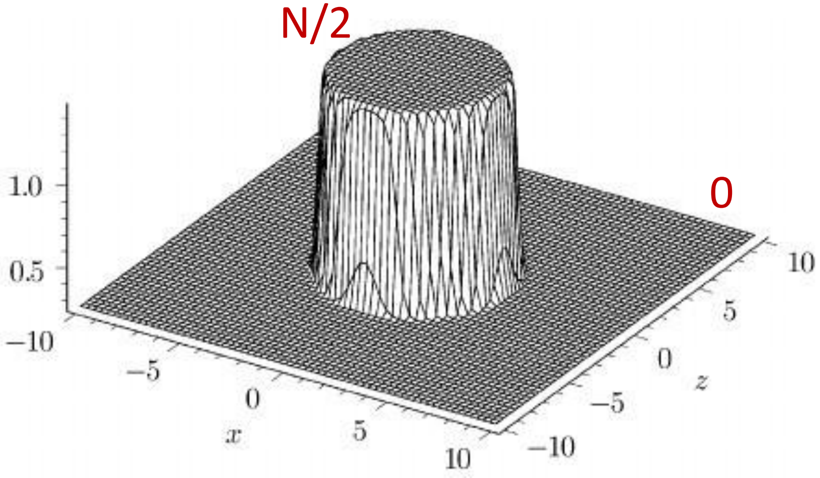


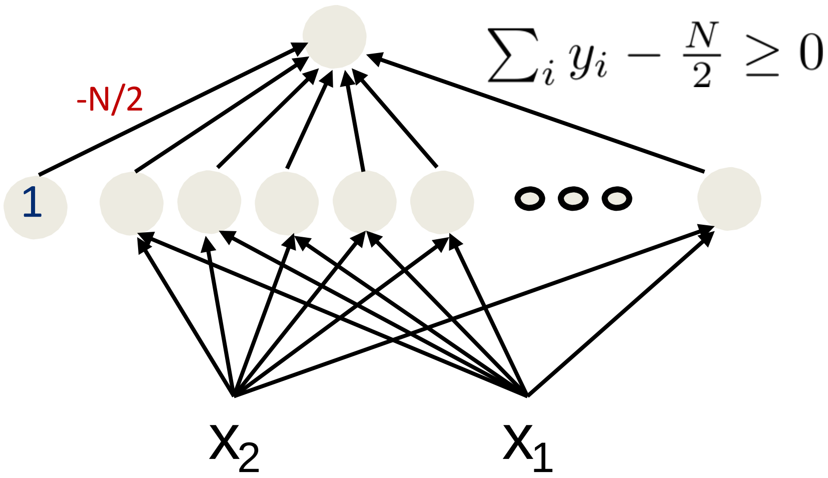
Composing a Circle

The circle network

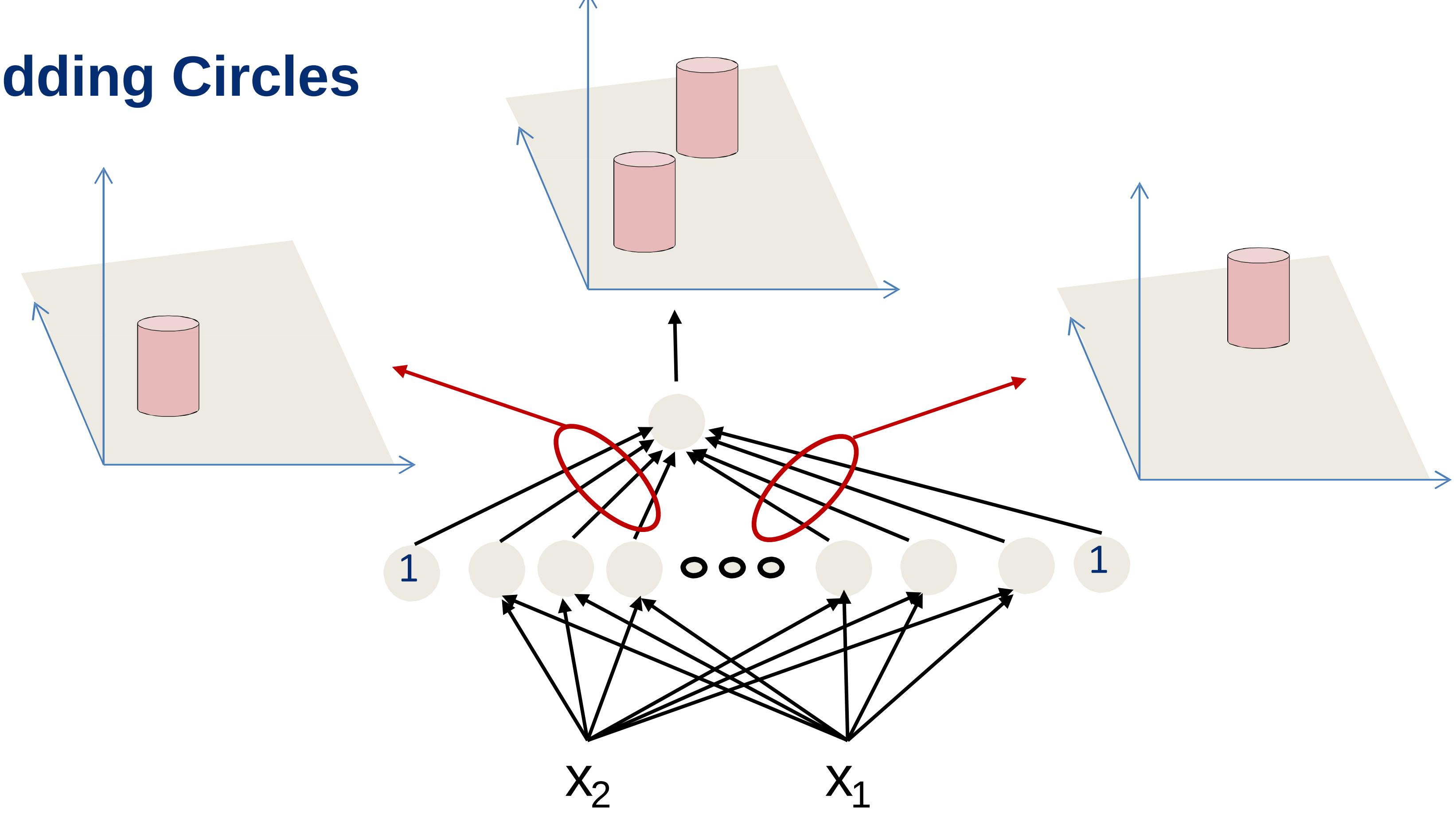
- Very large number of neurons (→ infinitely many)
- Sum is N inside circle & N/2 almost everywhere outside
- Circle can be at **any** location!







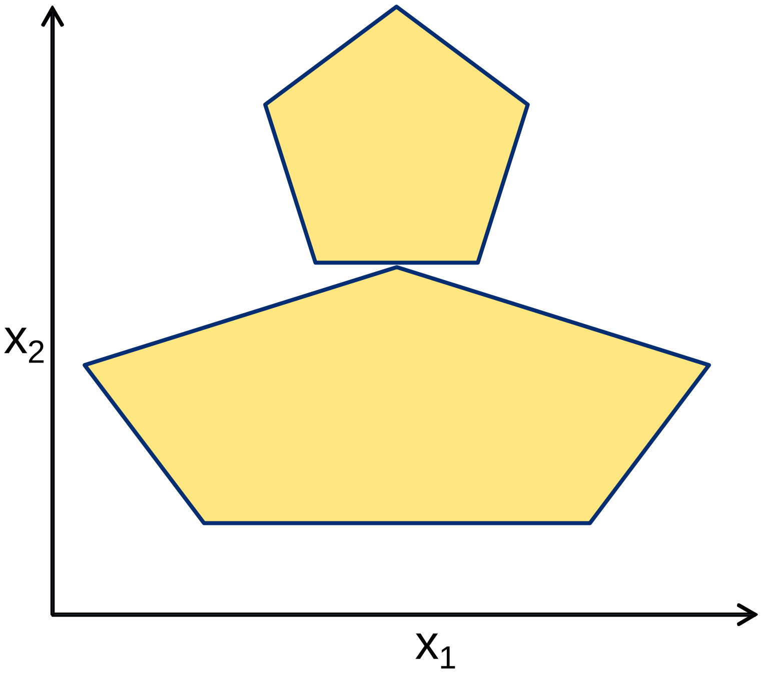
Adding Circles

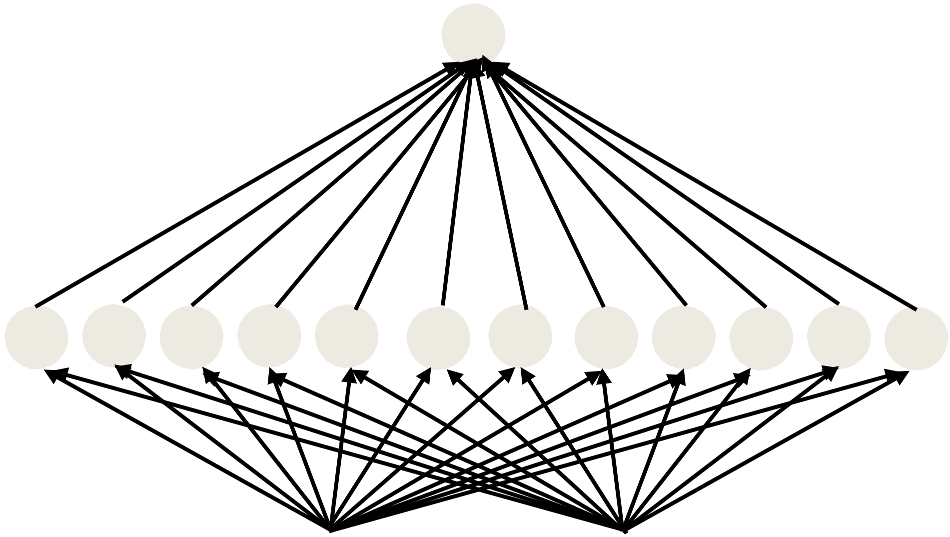


→ The "sum" of two circle sub-nets is exactly N/2 within circles and 0 almost everywhere else!

Even More Complex Decision Boundaries

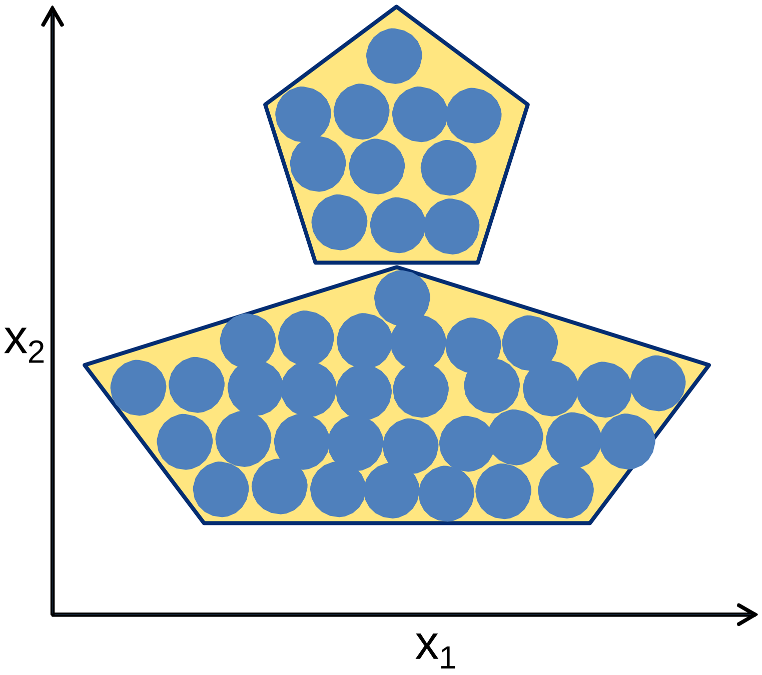
Task: Build network that fires in colored area **with only one hidden layer**

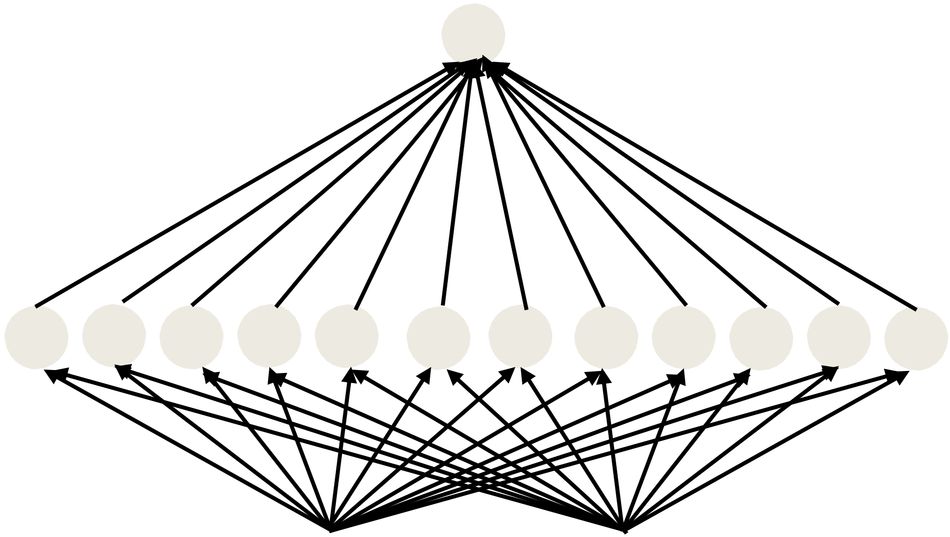




Even More Complex Decision Boundaries

Task: Build network that fires in colored area **with only one hidden layer**

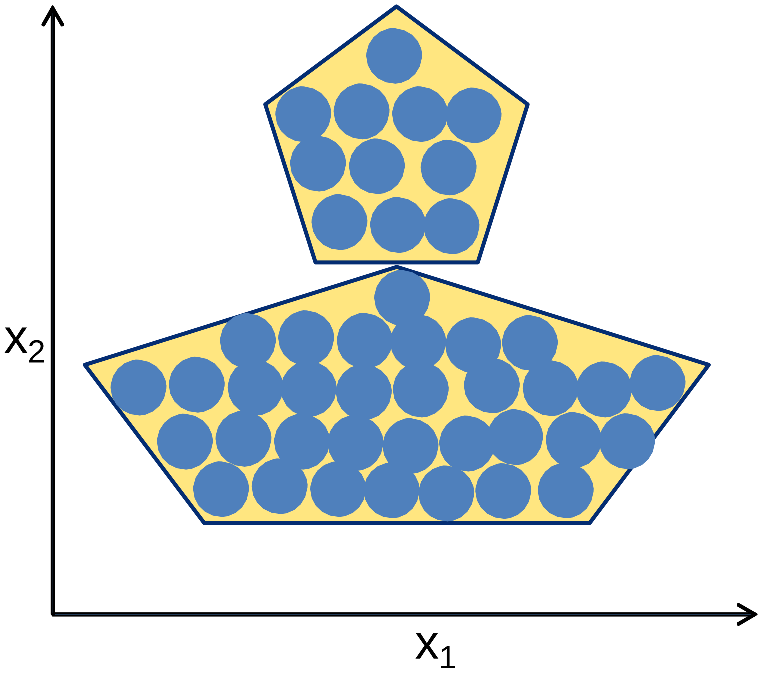


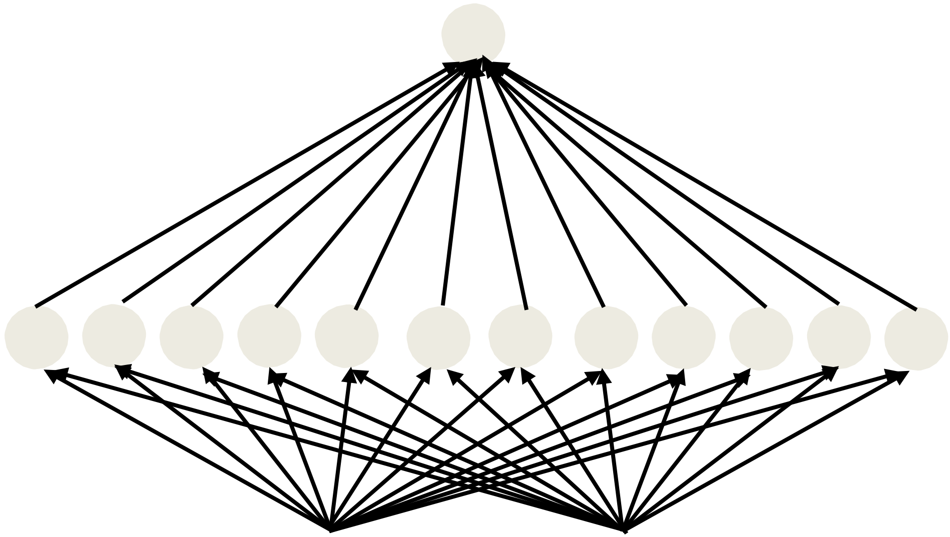


Even More Complex Decision Boundaries

A one-layer MLP can model ANY classification boundary!

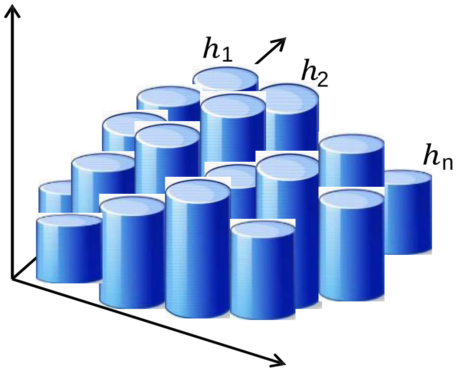
→ MLPs are universal classifiers!





…but it requires infinitely many neurons.

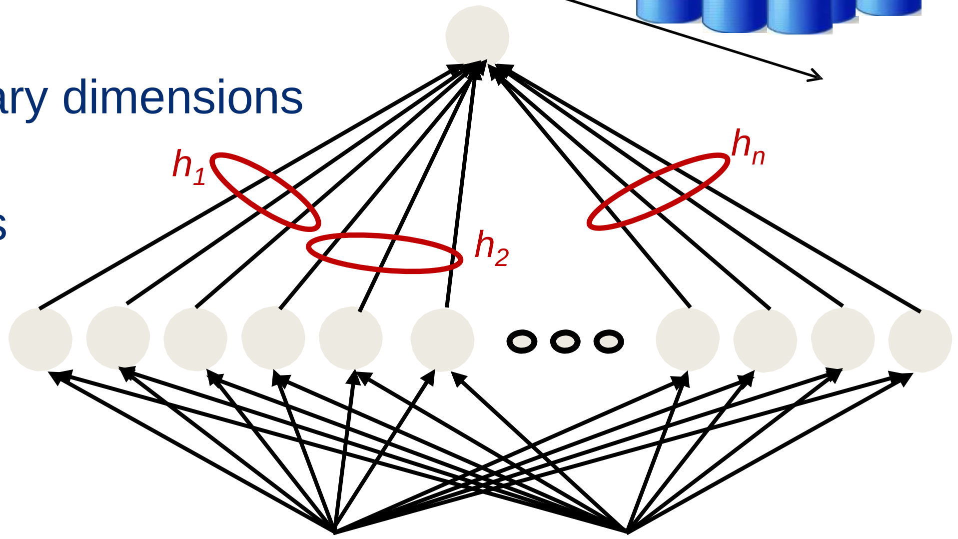
MLP as a continuous-valued function



Recall: Output of perceptron does not have to be Boolean
→ Can be regressor!

MLPs can compose functions in arbitrary dimensions

- This worked with even one layer!
  Sum of scaled and shifted cylinders

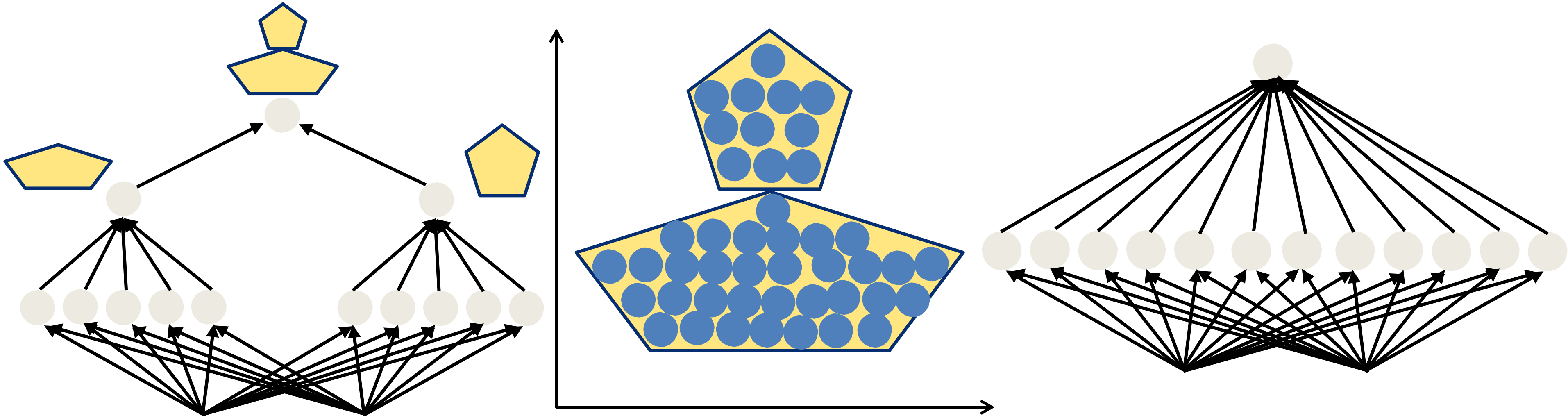


- Arbitrary precision
  → Thinner cylinders

→ MLP is a universal approximator!

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359-366.

The Need for Depth



Approximation with a single layer required infinitely many neurons!
→ Adding one hidden layer reduces this size to 13 neurons!

There is a lot of work on optimal depth, but we will not go into details here.

Introduction to and History of Neural Networks

Questions?