Neural Networks

1. Introduction

Spring 2023

Logistics: By now you must have…

- Already seen the logistics post on piazza
  - And possibly watched lecture 0 (logistics)
  - If not do so at once

- Been to the course website
  - http://deeplearning.cs.cmu.edu
  - If you have not done so, please visit it at once

- Course objectives, logistics, quiz and homework policies, and grading policies, all have been explained in both, the logistics lecture and on the course page

- Please familiarize yourself with this information at once

Logistics: Part 2

- You should already have
  - Signed on to piazza
  - Verified you have access to canvas and autolab
  - Ensured you have AWS accounts setup
    - And tested out Google colab

- You will get a note on forming study groups
  - We recommend this; you learn better in teams than you do by yourself
  - **Please sign up for the study groups *immediately!!!!!!!!!!***

Course philosophy and resources

- No student left behind : In our ideal world everyone of you would earn an A

- Please use the available resources
  - TAs
  - Study groups and TA mentors
    - Collaboration is encouraged
  - Dozens of office hours weekly
  - Me (email me, or just walk into my office if I'm free)
  - Your classmates and friends

- If under stress/unable to perform, please reach out
  - To your TA mentor
  - To me
  - We will do our best to help you

Hackathons

- **We will hold "hackathons" on Saturdays**
  - **Room: TBD**
  - For homework assignments
- You may join, with you study group
- There will be TAs monitoring/supporting
- You may discuss with other students/groups to help you with your assignments

- It is also a great place to meet fellow students and form study/project teams if you don't have one already

- The first Hackathon is this Saturday

Course Objectives: By the end of the course you will be able to…

1. Understand some of the theory behind neural networks
2. Build your own neural net components and tools
3. Work on large-scale problems

Course Objectives: By the end of the course you will be able to…

1. Understanding some of the theory behind neural networks
   - The what, the why and the how
     - The math
     - And the occasional history
   - Will help you contextualize 2 and 3 below
   - Will help you develop and extend your ideas in the topic
     - Research / Grad school
     - And job interviews!

2. Build your own neural net components and tools
3. Work on large-scale problems

Course Objectives: By the end of the course you will be able to…

1. Understanding some of the theory behind neural networks
2. Build your own neural net components and tools
   - Part 1s of your homeworks
   - Bonus problems
3. Work on large-scale problems

Course Objectives: By the end of the course you will be able to…

1. Understanding some of the theory behind neural networks
2. Build your own neural net components and tools
3. Work on large-scale problems
   – Part 2s of your homeworks

Course Objectives: By the end of the course you will be able to…

1. Understanding some of the theory behind neural networks
2. Build your own neural net components and tools
3. Work on large-scale problems

- Course projects may relate to 1, 2 or 3

Lecture Style

- My lecture style is verbose, with lots of visualization
- Many many slides
  - With a lot of animation
- Given a choice between deriving an equation symbolically, and explaining it with 30 slides of pictures and animation, I usually choose the latter
- If this is not your cup of tea, this is not the class for you

Attendance

- We will use in-class polls to verify attendance
  - Multiple polls posted at random times through the class
  - Polls will be posted on piazza
    - Please keep your piazza (and *only* your piazza) open
  - You must respond to all polls
    - We don’t score you on correctness, only on whether you responded

- Students who have permission to view videos instead: **please watch mediatech videos**
  - We will gather your attendance from there

Classroom Engagement

- This is an interactive class
- We like questions
  - No question is silly/wrong/embarrassing
  - If you have a question, odds are that others have it too
- We will try different tactics to encourage (or enforce) interaction
  - E.g. you will be given numbers
  - When your number is called, you must answer the question asked
    - "I don't know" is an acceptable answer
    - Some of the questions will be "socratic"
      - Without clear answers, intended to make you think
- Please participate

Classroom Engagement: 2

- This is a "dumb" classroom
  - Keep your smart devices shut
  - No phones or laptops open except for the following:
- Exceptions:
  - To answer polls
  - To view lecture slides
  - To take notes
  - If we find you using your devices for any other purpose, we may ask you to leave the room

I’m handicapped

- I’m physically handicapped
- My back and neck are bent. I cannot raise my arms. On some days I cannot walk. My breath runs out midway through the lecture

- Please do not mind my mannerisms
  – I may ask my TAs to clip on my mic
    - I cannot do so myself
  – I use strange devices to point, because they are light
    - I cannot raise my arm enough to point

A minute for questions…

Caveat: Slide deck often have many "hidden" slides that will not be shown during the lecture, but will feature in your weekly quizzes

Today’s lessons

- A brief history of neural networks
  - Connectionism
    - Its relation to cognition and the brain
    - Its contrast to conventional computer architecture
  - Early models, and their limitations

- Introducing modern neural networks

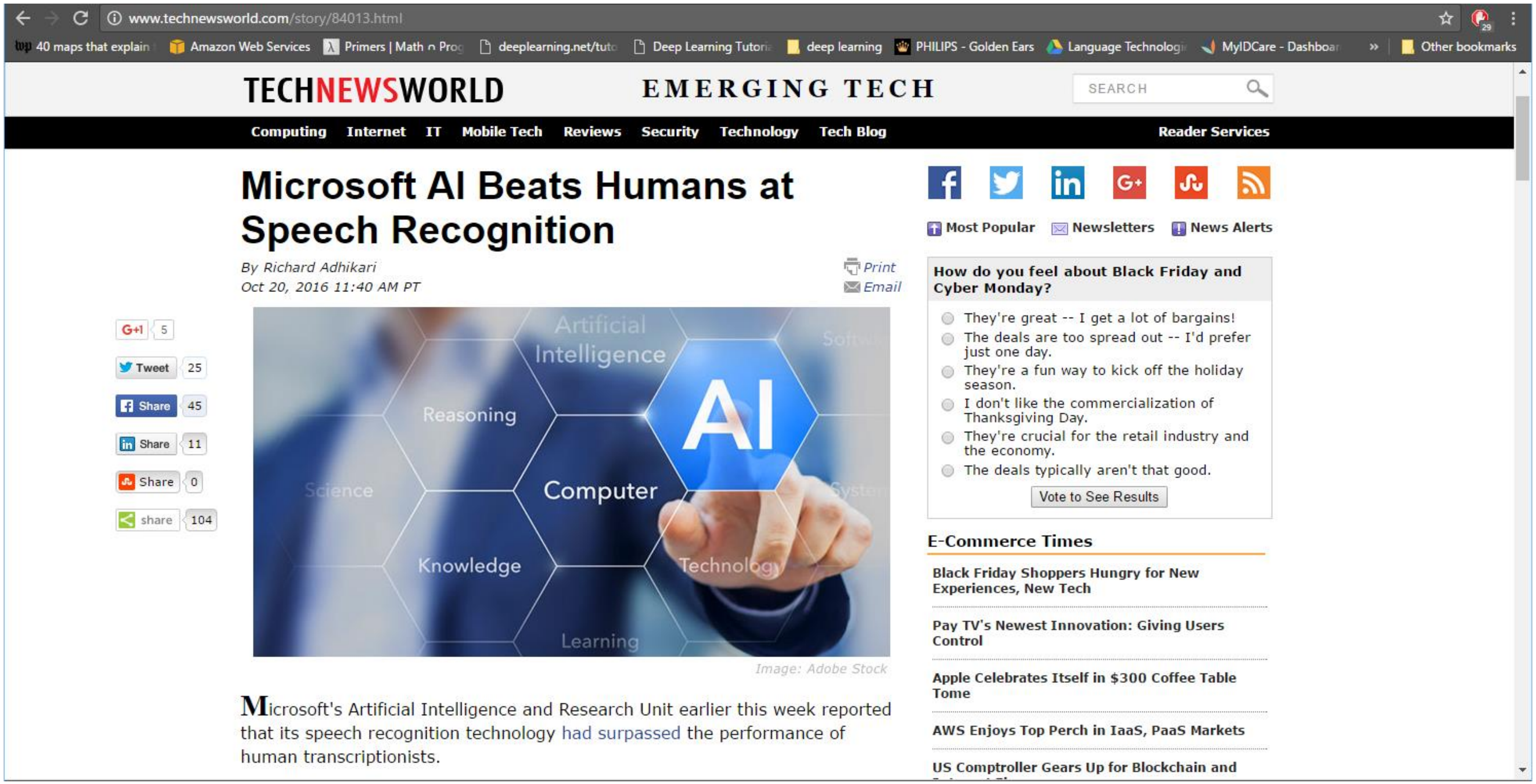
- And what they can compute

Neural Networks are taking over!

- Neural networks have become one of *the* main approaches to AI

- They have been successfully applied to various pattern recognition, prediction, and analysis problems
- In many problems they have established the state of the art
  - Often exceeding previous benchmarks by large margins
  - Sometimes solving problems you couldn't solve using earlier ML methods

Breakthroughs with neural networks



Breakthrough with neural networks

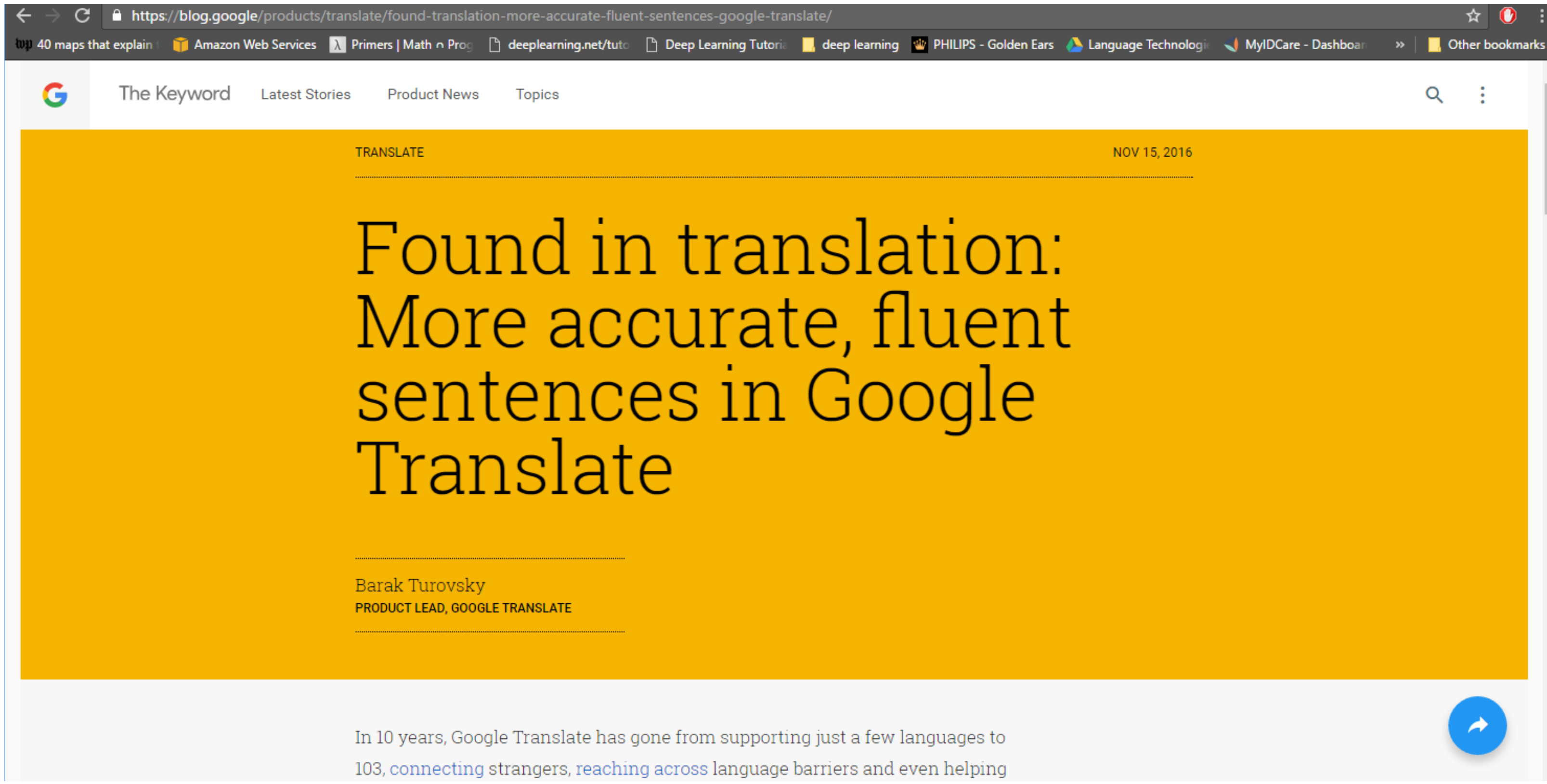
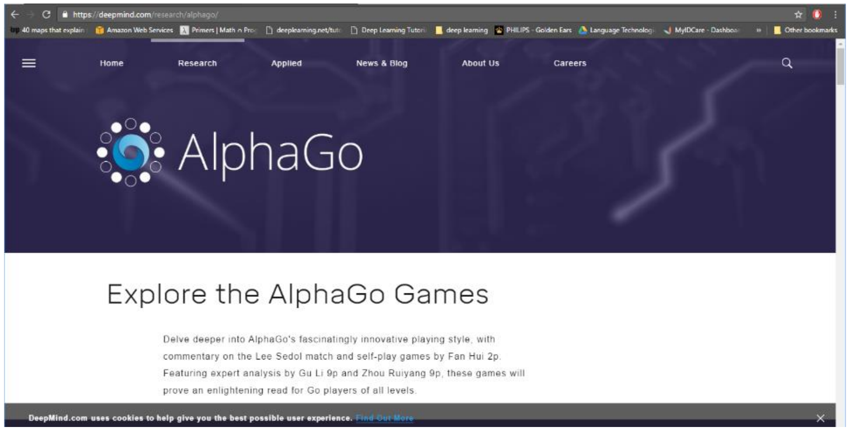


Image segmentation and recognition

Breakthroughs with neural networks



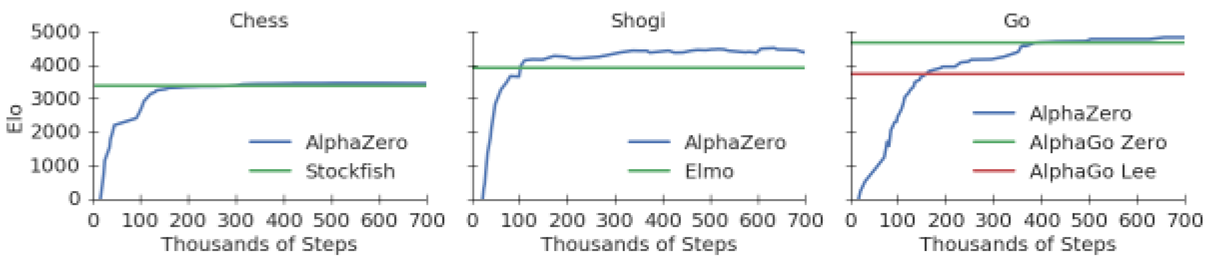
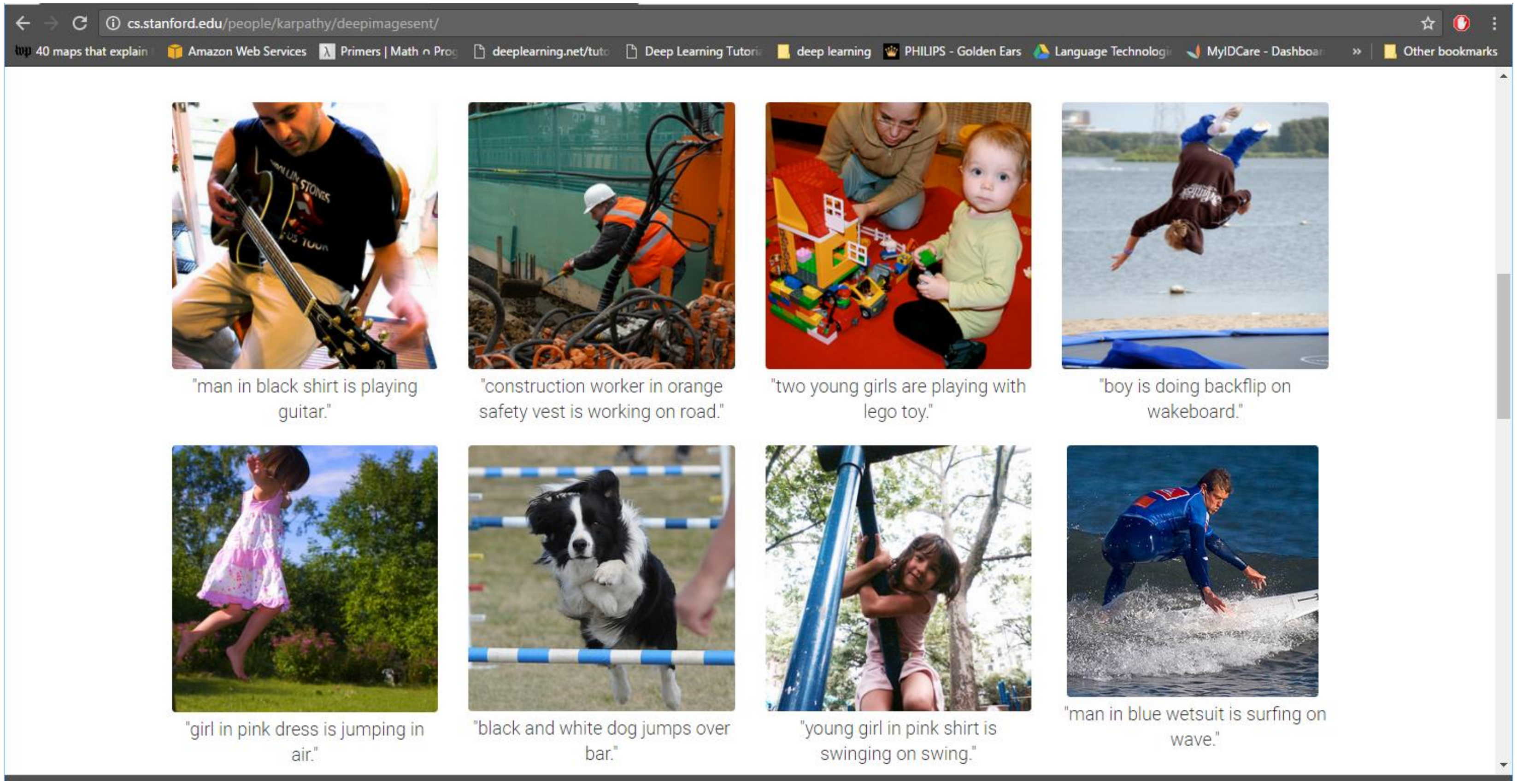


Figure 1: Training *AlphaZero* for 700,000 steps. Elo ratings were computed from evaluation games between different players when given one second per move. **a** Performance of *AlphaZero* in chess, compared to 2016 TCEC world-champion program *Stockfish*. **b** Performance of *AlphaZero* in shogi, compared to 2017 CSA world-champion program *Elmo*. **c** Performance of *AlphaZero* in Go, compared to *AlphaGo Lee* and *AlphaGo Zero* (20 block / 3 day) (*29*).

Success with neural networks



- Captions generated entirely by a neural network

Breakthroughs with neural networks

DALL-E 2

"A bowl of soup that looks like a monster made of play dough" according with Dall-E. Image: OpenAI

ChatGPT

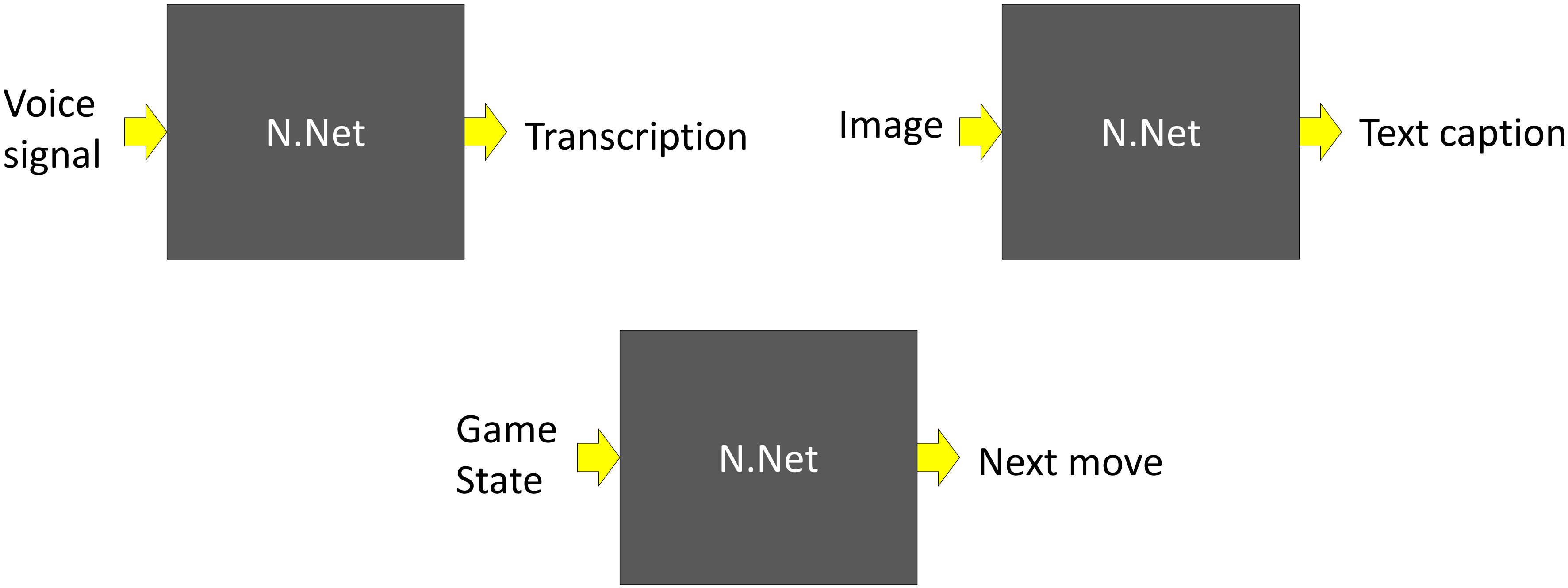
BH tell me the story of perseus and andromeda in iambic pentameter

Perseus, a brave and strong young man,
Whose father was a god, set out one day
To rescue fair Andromeda, who ran
Afar to escape her fate of being prey
To Cetus, monster of the ocean spray.
With winged sandals and a mirrored shield,
Perseus flew to where the damsel lay
Chained to a rock and waiting to be killed.
He fought and slayed the beast, and set her free,
And brought her back to Greece, to be his bride.
Their love was strong and true, and all could see
The gods had blessed their union far and wide.
Thus Perseus and Andromeda's names,
Were etched in fame for all immortal games.

Successes with neural networks

- And a variety of other problems:
  - From art to astronomy to healthcare..
  - and even predicting stock markets!

So what are neural networks??

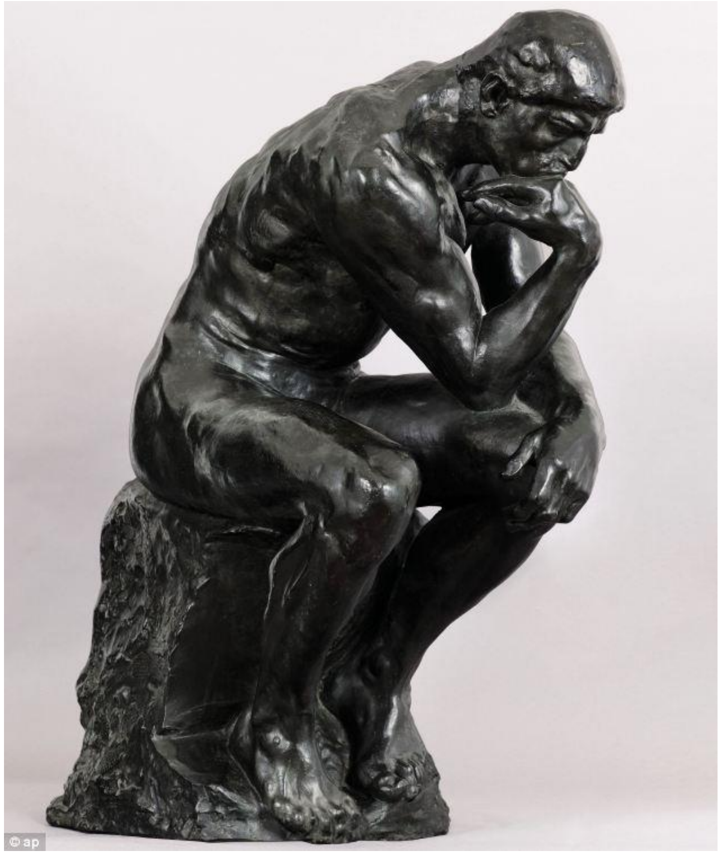


- What are these boxes?

So what are neural networks??

- It begins with this..

So what are neural networks??



"The Thinker!"
by Augustin Rodin

- Or even earlier.. with this..

The magical capacity of humans

- Humans can
  - Learn
  - Solve problems
  - Recognize patterns
  - Create
  - Cogitate
  - …

Dante!

- Worthy of emulation
- But how do humans “work“?

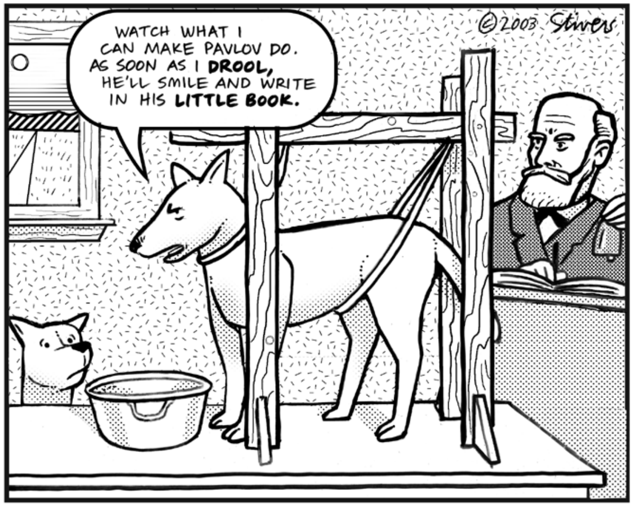
Cognition and the brain..

- "If the brain was simple enough to be understood - we would be too simple to understand it!"
  – Marvin Minsky

Early Models of Human Cognition

- Associationism
  - Humans learn through association
- 400BC-1900AD: Plato, David Hume, Ivan Pavlov..

What are "Associations"



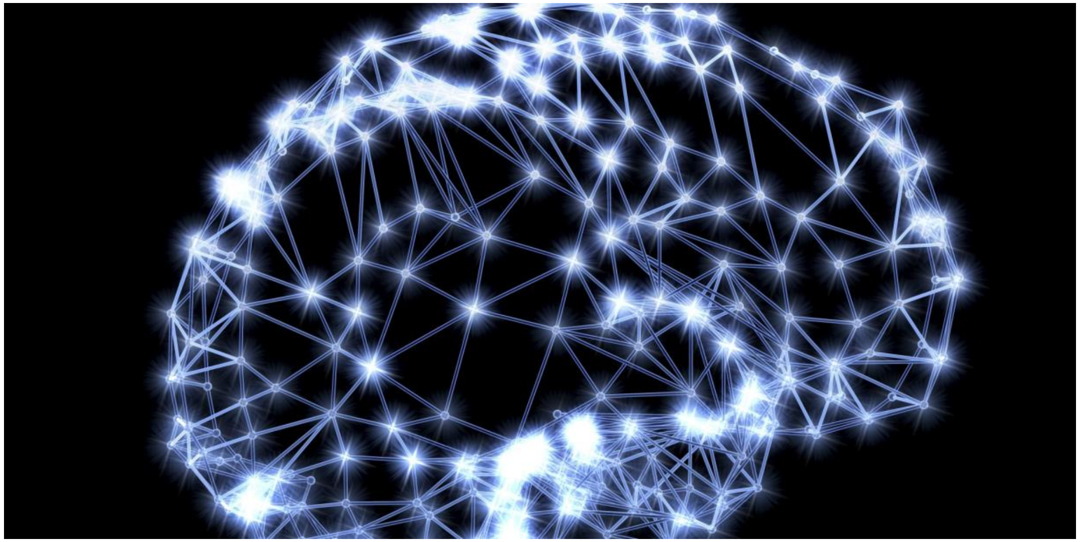
- Lightning is generally followed by thunder
  - Ergo – "hey here's a bolt of lightning, we're going to hear thunder"
  - Ergo – "We just heard thunder; did someone get hit by lightning"?

- Association!

- **But where are the associations stored??**

- **And how?**

Observation: ***The Brain***

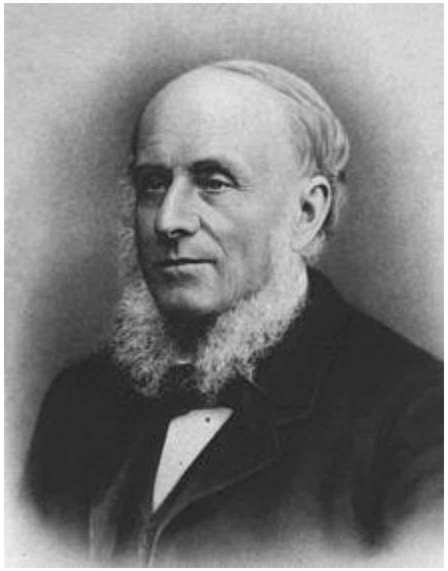


- Mid 1800s: The brain is a mass of interconnected neurons

Brain: Interconnected Neurons

- Many neurons connect *in* to each neuron
- Each neuron connects *out* to many neurons

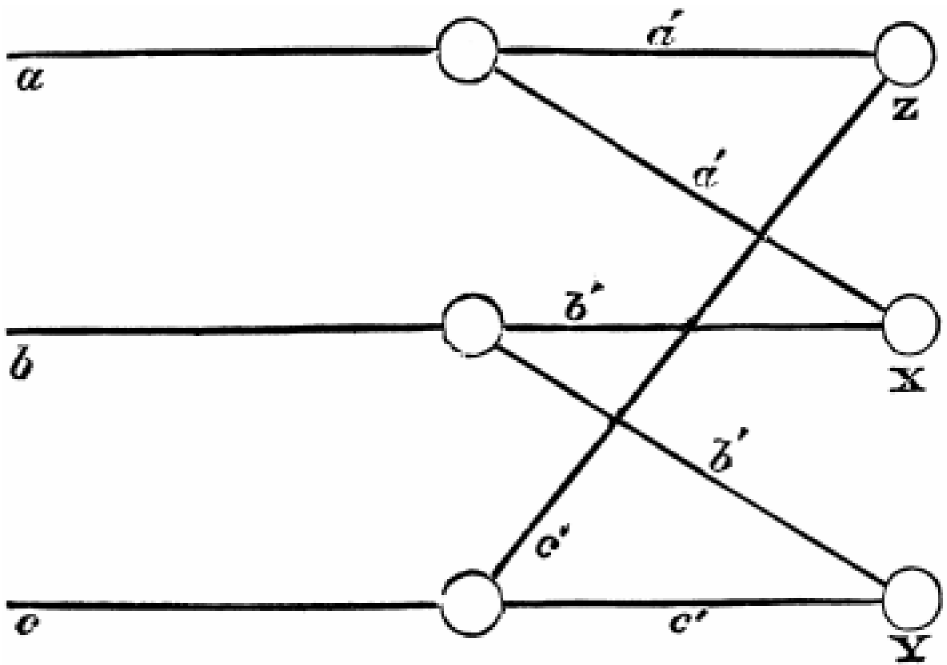
Enter ***Connectionism***



- Alexander Bain, philosopher, psychologist, mathematician, logician, linguist, professor
- 1873: The information is in the ***connections***
  - *Mind and body* (1873)

Bain’s Idea 1: Neural Groupings

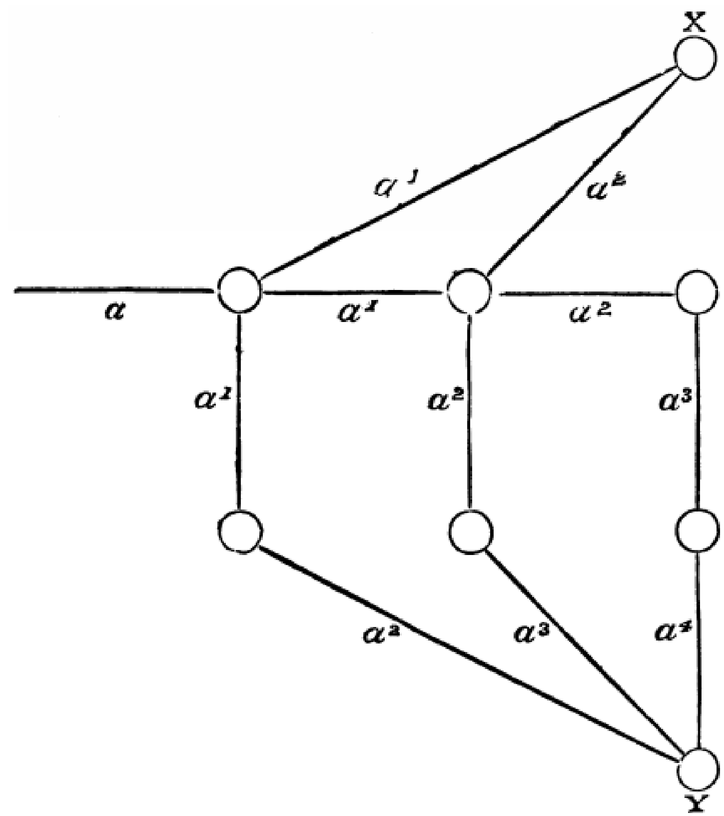
- Neurons excite and stimulate each other
- Different combinations of inputs can result in different outputs



Bain's Idea 1: Neural Groupings

- Different intensities of activation of A lead to the differences in when X and Y are activated

- Even proposed a learning mechanism..



Bain's Idea 2: Making Memories

- "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."

- Predicts "Hebbian" learning (three quarters of a century before Hebb!)

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

- In 1873, Bain postulated that there must be one million neurons and 5 billion connections relating to 200,000 "acquisitions"

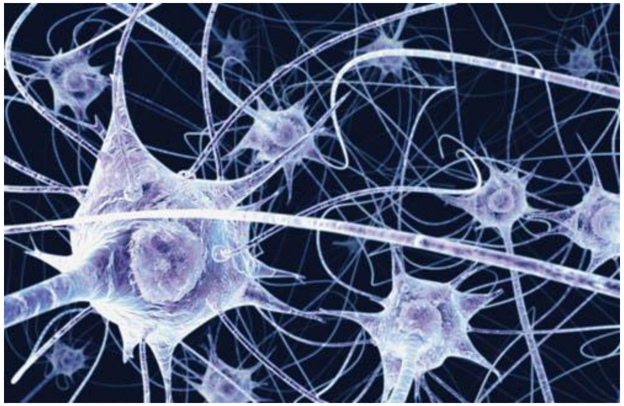
- In 1883, Bain was concerned that he hadn't taken into account the number of "partially formed associations" and the number of neurons responsible for recall/learning

- By the end of his life (1903), recanted all his ideas!
  - Too complex; the brain would need too many neurons and connections

Connectionism lives on..

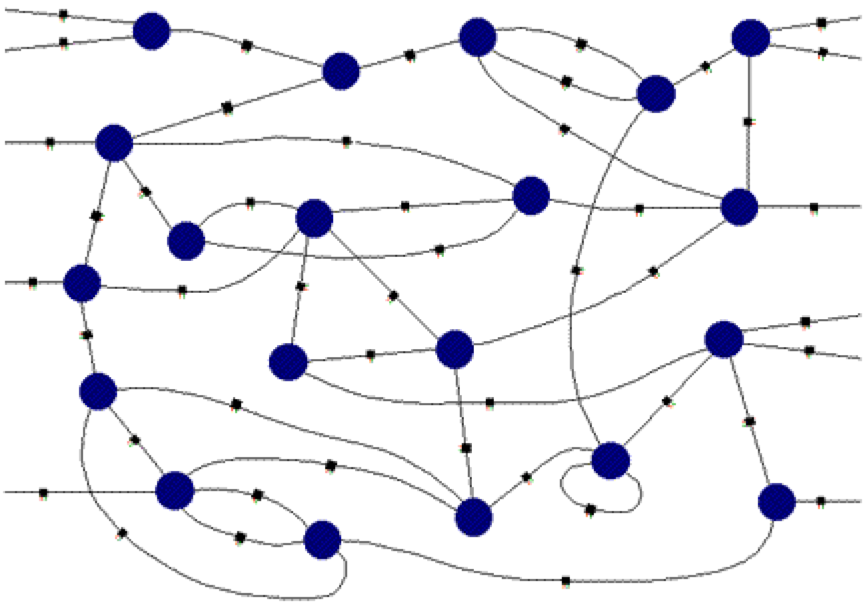
- The human brain is a connectionist machine
  - Bain, A. (1873). *Mind and body. The theories of their relation.* London: Henry King.
  - Ferrier, D. (1876). *The Functions of the Brain.* London: Smith, Elder and Co

- Neurons connect to other neurons. The processing/capacity of the brain is a function of these connections



- Connectionist machines emulate this structure

Connectionist Machines



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

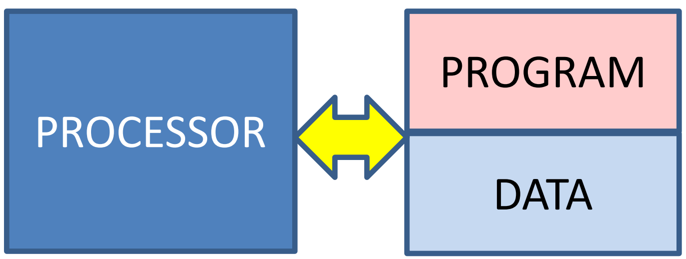
Connectionist Machines

→ connections are the program, very hard to build so we emulate

- Neural networks are *connectionist* machines
  - As opposed to Von Neumann Machines

→ Processors are seperate from program & data

Von Neumann/Princeton Machine



Processing unit

Memory

Neural Network

We emulate connectionist machines after Von Neumann Machines by writing Programs for it.

- The machine has many non-linear processing units
  - The program is the connections between these units
    - Connections may also define memory

Recap

- Neural network based AI has taken over most AI tasks
- Neural networks originally began as computational models of the brain
  - Or more generally, models of cognition
- The earliest model of cognition was *associationism*
- The more recent model of the brain is *connectionist*
  - Neurons connect to neurons
  - The workings of the brain are encoded in these connections
- Current neural network models are *connectionist machines*

Poll 1

1. Who is the first person that proposed connectionism? (Single Choice):

- Aristotle
- Alexander Bain
- David Hartley
- Alan Turing

2. Roughly how many connections exist between neurons in the brain? (Single Choice):

- 1 million
- 5 billion
- 80 billion
- 100 trillion → ChatGPT has 175B connections, GPT4 will have 1T conn?

Poll 1

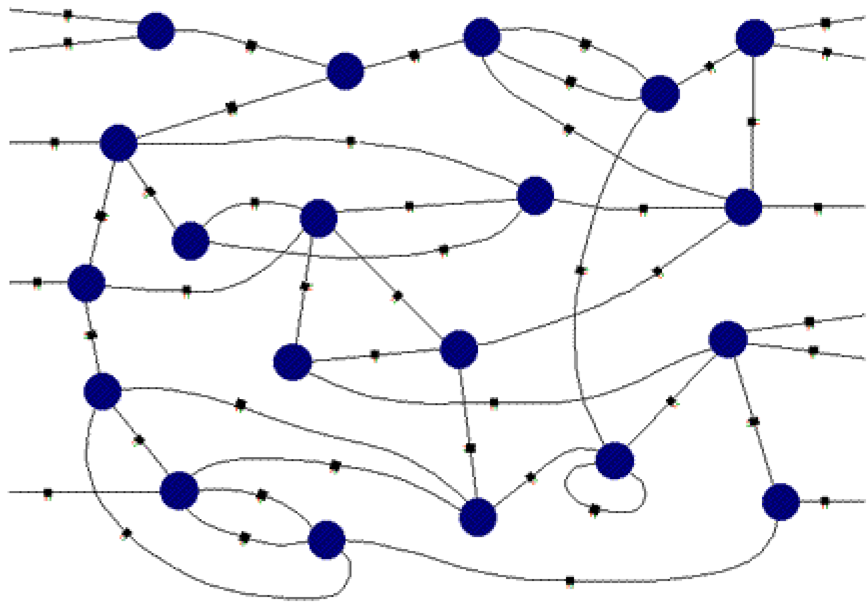
1. Who is the first person that proposed connectionism? (Single Choice):

- Aristotle
- **Alexander Bain**
- David Hartley
- Alan Turing

2. Roughly how many connections exist between neurons in the brain? (Single Choice):

- 1 million
- 5 billion
- 80 billion
- **100 trillion**

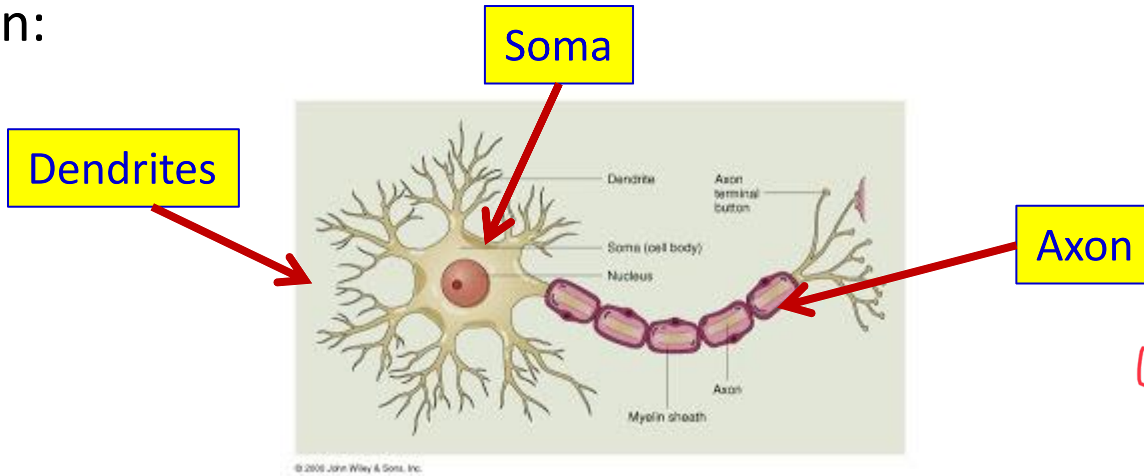
Connectionist Machines



- Network of processing elements
  - All world knowledge is stored in the *connections* between the elements
- *But what are the individual elements?*

Modelling the brain

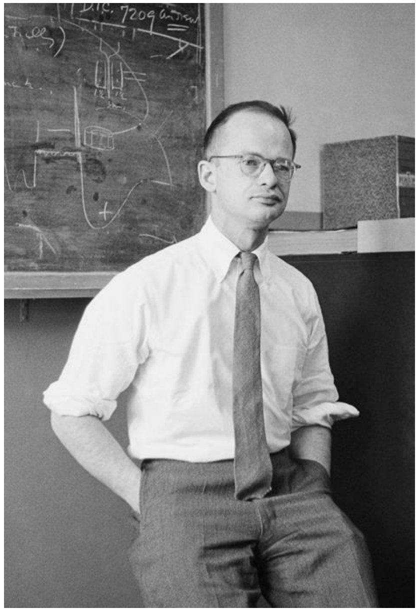
- What are the units?
- A neuron:

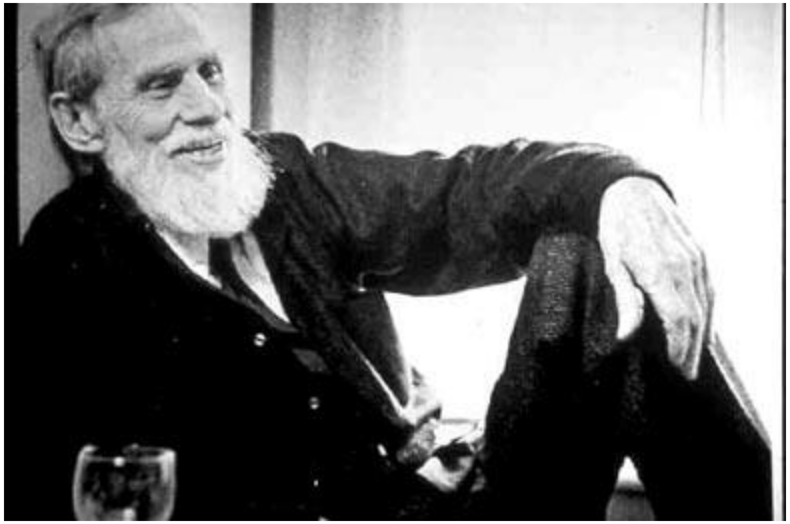


(protected by Fat,
Myelin Sheath

- Signals come in through the dendrites into the Soma
- A signal goes out via the axon to other neurons
  - Only one axon per neuron
- Factoid that may only interest me: Neurons do not undergo cell division
  - Neurogenesis occurs from neuronal stem cells, and is minimal after birth

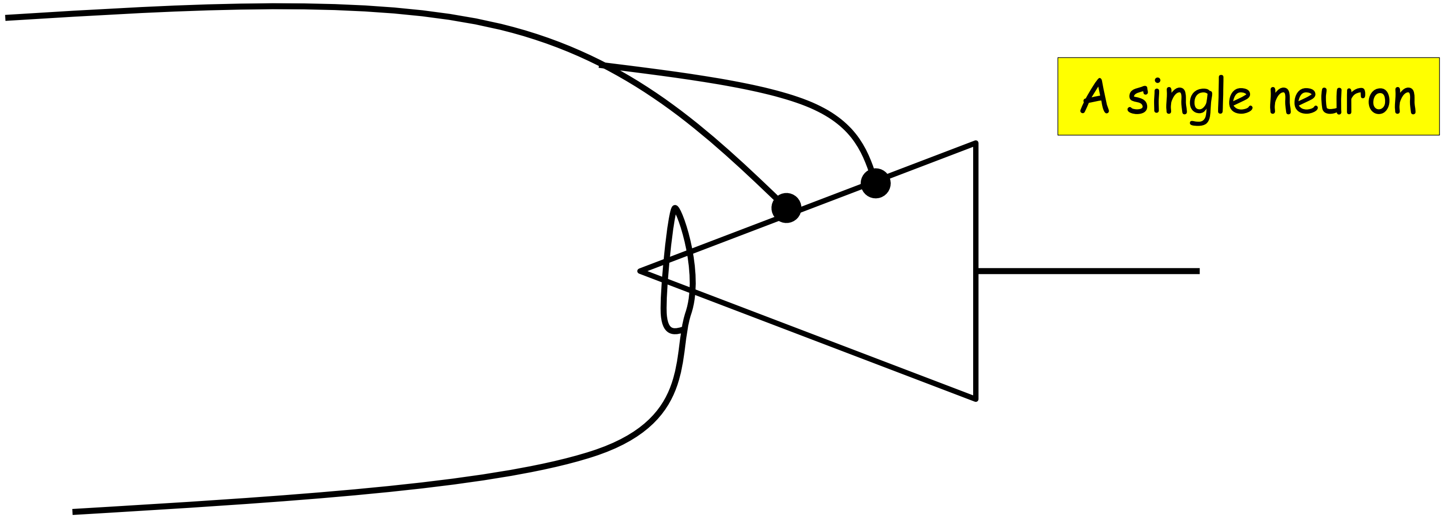
McCulloch and Pitts





- The Doctor and the Hobo..
  - Warren McCulloch: Neurophysiologist
  - Walter Pitts: Homeless wannabe logician who arrived at his door

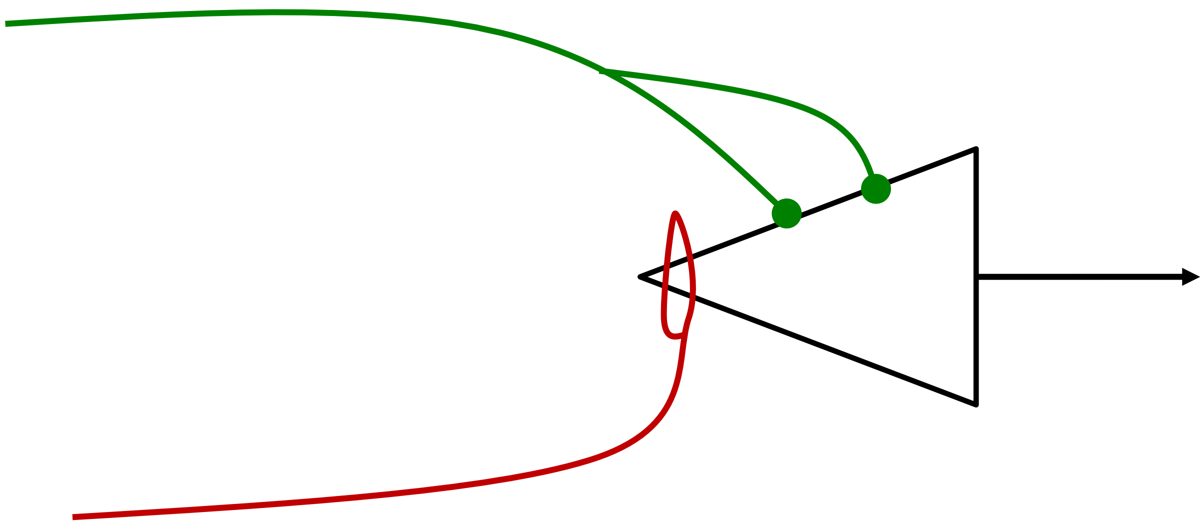
The McCulloch and Pitts model



Func of brain can be explained via boolean logic.

- A mathematical model of a neuron
  - McCulloch, W.S. & Pitts, W.H. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics, 5:115-137, 1943
    - Pitts was only 20 years old at this time

Synaptic Model

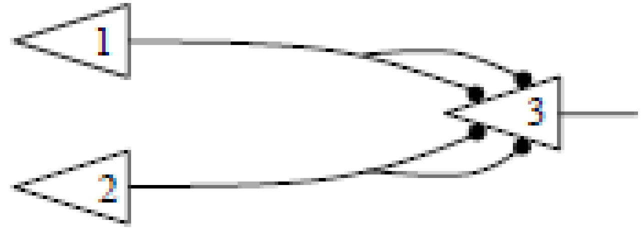


- ***Excitatory synapse:*** Transmits weighted input to the neuron if threshold is exceeded
- ***Inhibitory synapse:*** Any signal from an inhibitory synapse prevents neuron from firing
  - The activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time.
    - Regardless of other inputs

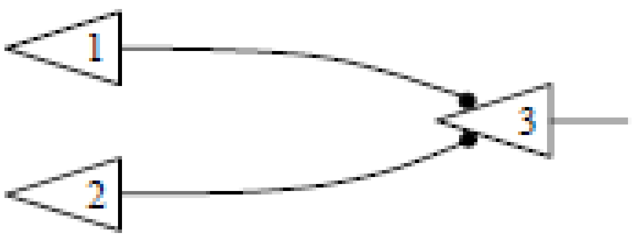
Simple "networks" of neurons can perform Boolean operations

Boolean Gates

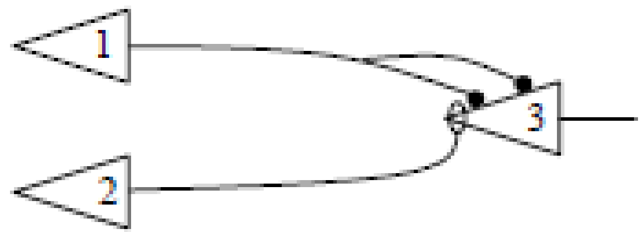
$N_2(t)) \leftrightarrow N_1(t\text{-}1)$
net for temporal predecessor



$N_3(t) \leftrightarrow N_1(t\text{-}1) \vee N_2(t\text{-}1)$
net for disjunction



$N_3(t) \leftrightarrow N_1(t\text{-}1) \& N_2(t\text{-}1)$
net for conjunction



$N_3(t) \leftrightarrow N_1(t\text{-}1) \& \sim N_2(t\text{-}1)$
net for conjunction and negation

Figure 1. Diagrams of McCulloch and Pitts nets. In order to send an output pulse, each neuron must receive two excitory inputs and no inhibitory inputs. Lines ending in a dot represent excitatory connections; lines ending in a hoop represent inhibitory connections.

Complex Percepts & Inhibition in action

They can even create illusions of "perception"

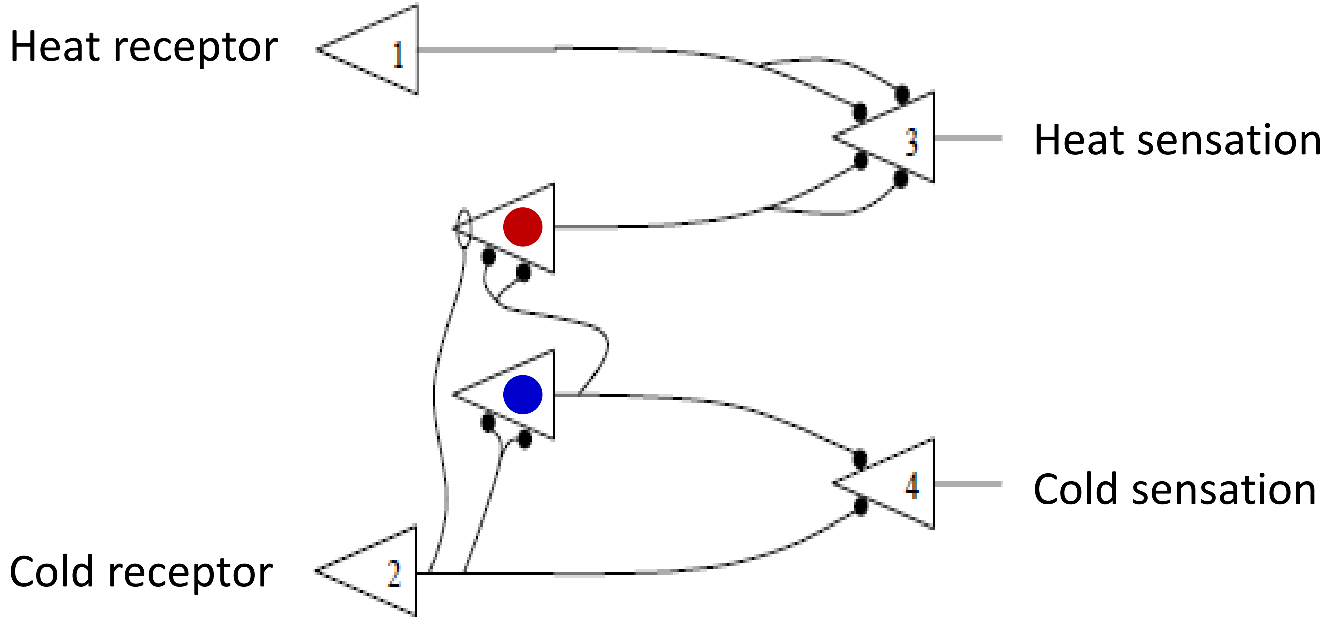


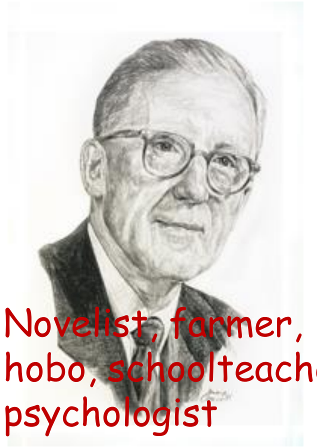
Figure 2. Net explaining the heat illusion. Neuron 3 (heat sensation) fires if and only if it receives two inputs, represented by the lines terminating on its body. This happens when either neuron 1 (heat reception) fires or neuron 2 (cold reception) fires once and then immediately stops firing. When neuron 2 fires twice in a row, the intermediate (unnumbered) neurons excite neuron 4 rather than neuron 3, generating a sensation of cold.

Criticisms

- They claimed that their nets
  - Should be able to compute a small class of functions
  - Also, if tape is provided their nets can compute a richer class of functions.
    - They will be equivalent to Turing machines
    - Claim that they're Turing complete
  - They didn't prove any results themselves
- Didn't provide a learning mechanism..

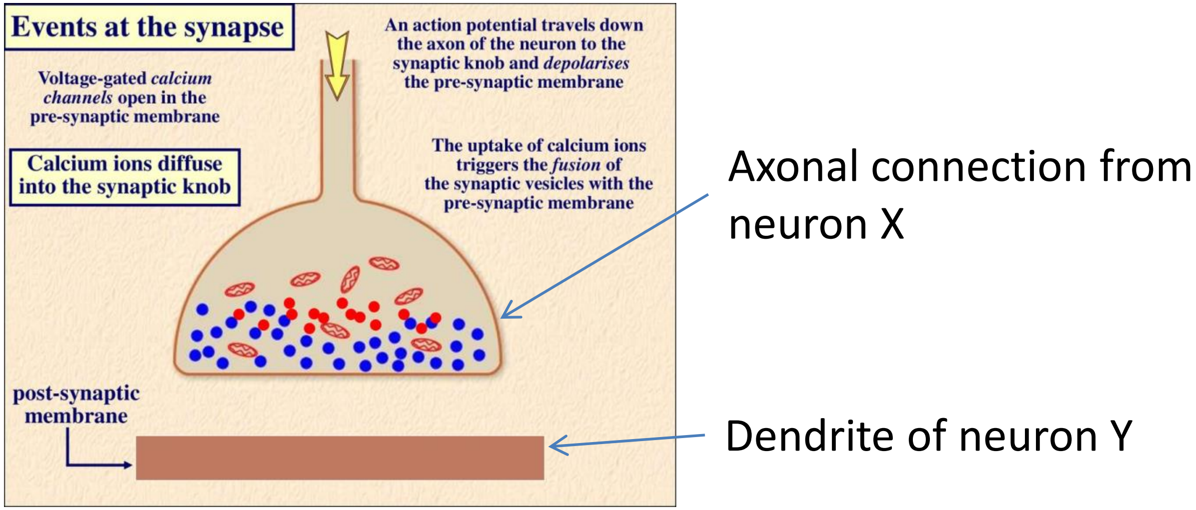
Father of Modern NN

Donald Hebb



- "Organization of behavior", 1949
- 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*

Hebbian Learning



- If neuron x repeatedly triggers neuron y, the synaptic knob connecting x to y gets larger
- In a mathematical model:

$$w_{xy} = w_{xy} + \eta xy$$

  - Weight of the connection from input neuron x to output neuron y
- This simple formula is actually the basis of many learning algorithms in ML

Problem? Weights do not decrease.

Hebbian Learning

- **Fundamentally unstable**
  - Stronger connections will enforce themselves
  - No notion of "competition"
  - No *reduction* in weights
  - Learning is unbounded

- Number of later modifications, allowing for weight normalization, forgetting etc.
  - E.g. Generalized Hebbian learning, aka Sanger's rule

$$w_{ij} = w_{ij} + \eta y_j \left(x_i - \sum_{k=1}^{j} w_{ik} y_k \right)$$

?

  - The contribution of an input is incrementally *distributed* over multiple outputs..

Poll 2

Hebbian learning is… (Single Choice)

- Fundamentally stable since stronger connections will enforce themselves
- Fundamentally unstable since there is no reduction in weights
- Fundamentally stable since learning is unbounded
- Fundamentally unstable since weights compete for adjustment

Poll 2

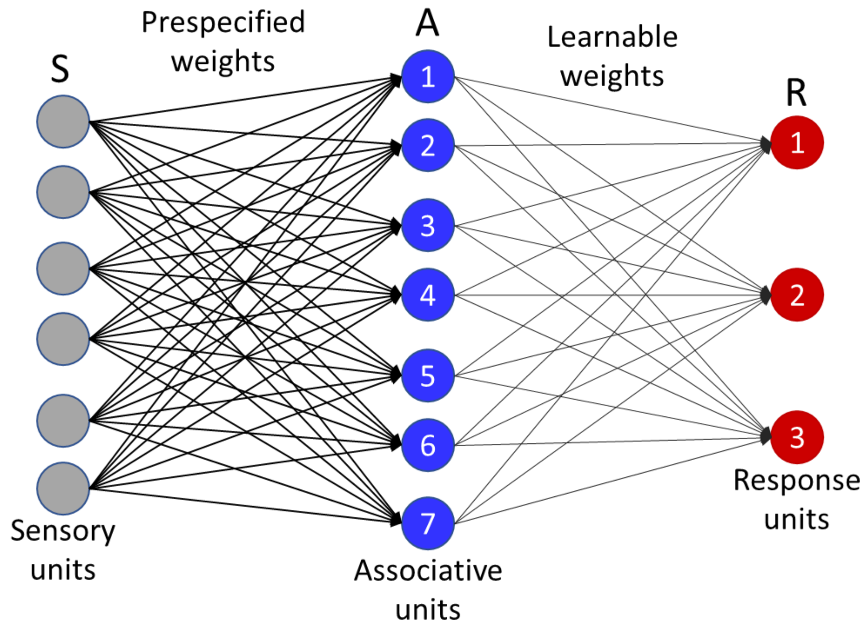
Hebbian learning is… (Single Choice)

- Fundamentally stable since stronger connections will enforce themselves
- **Fundamentally unstable since there is no reduction in weights**
- Fundamentally stable since learning is unbounded
- Fundamentally unstable since weights compete for adjustment

A better model

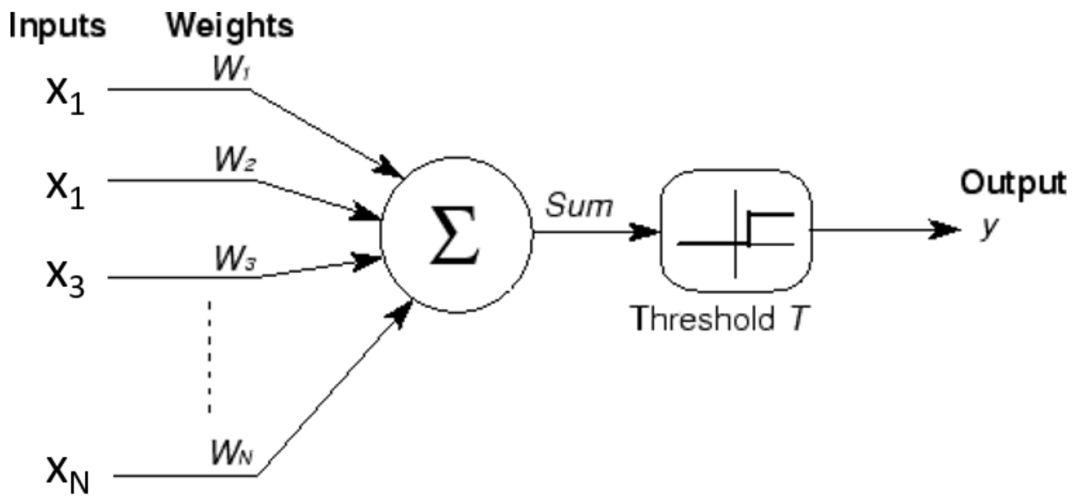
- Frank Rosenblatt
  - Psychologist, Logician
  - Inventor of the solution to everything, aka the Perceptron (1958)

Rosenblatt's perceptron



- Simplified perceptron model
  - Association units combine sensory input with fixed weights
  - Response units combine associative units with learnable weights

Perceptron: Simplified model



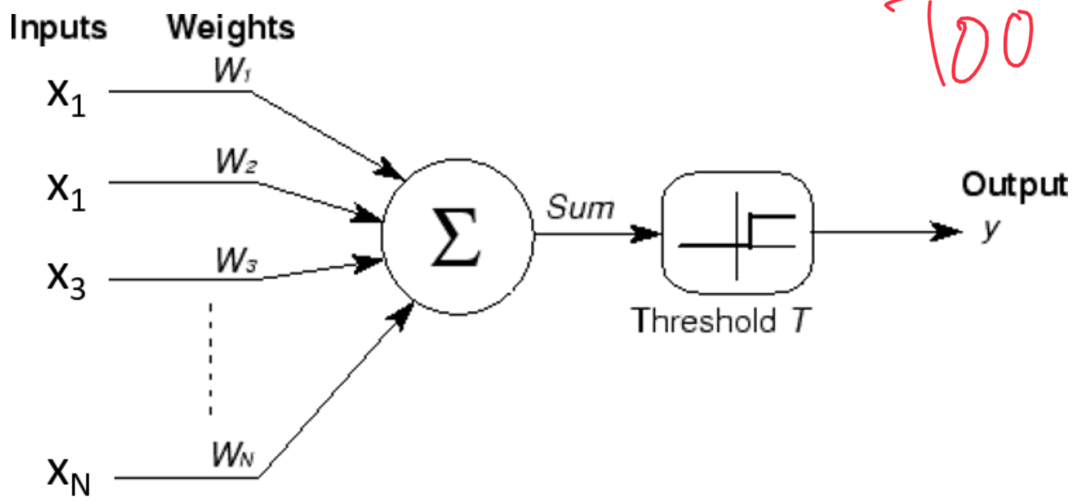
- Number of inputs combine linearly
  - Threshold logic: Fire if combined input exceeds threshold

$$Y = \begin{cases} 1 \;\; if \; \sum_i w_i x_i - T \geq 0 \\ 0 \qquad\qquad\qquad else \end{cases}$$

The Universal Model

- Originally assumed could represent *any* Boolean circuit and perform any logic
  - *"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,"* New York Times (8 July) 1958
  - *"Frankenstein Monster Designed by Navy That Thinks,"* Tulsa, Oklahoma Times 1958

TOO HYPE!



Also provided a learning algorithm

Target o/p

$$\mathbf{w} = \mathbf{w} + \eta(d(\mathbf{x}) - y(\mathbf{x}))\mathbf{x}$$

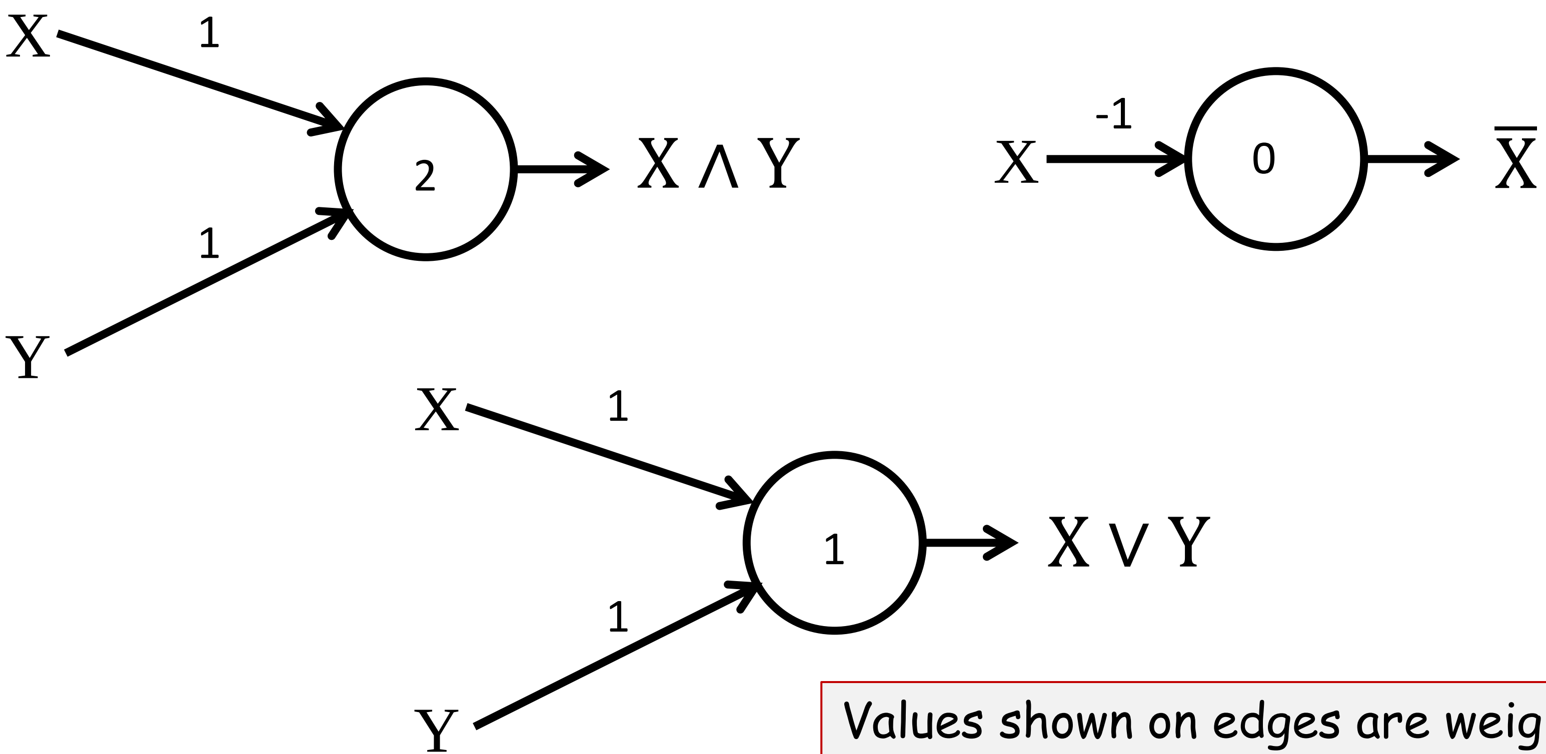
Sequential Learning:

$d(x)$ is the desired output in response to input $\mathbf{x}$

$y(x)$ is the actual output in response to $\mathbf{x}$

- Boolean tasks
- Update the weights whenever the perceptron output is wrong
  - Update the weight by the product of the input and the *error* between the desired and actual outputs
- Proved convergence for linearly separable classes

Perceptron

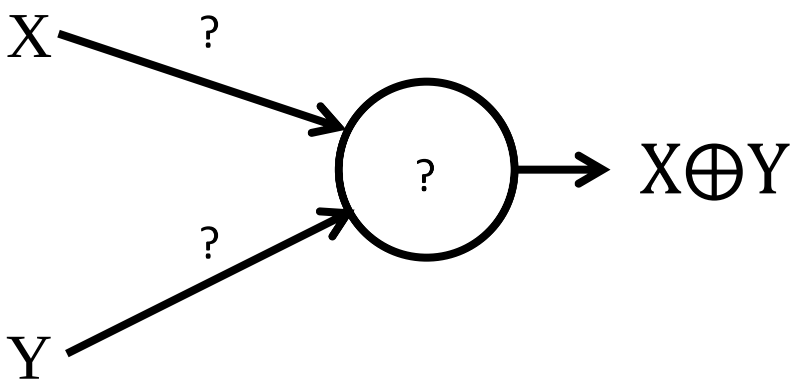


Values shown on edges are weights, numbers in the circles are thresholds

- Easily shown to mimic any Boolean gate
- But…

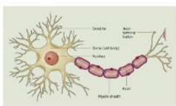
Individual units

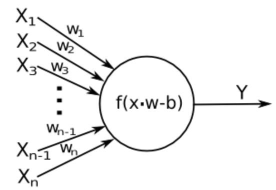
No solution for XOR!

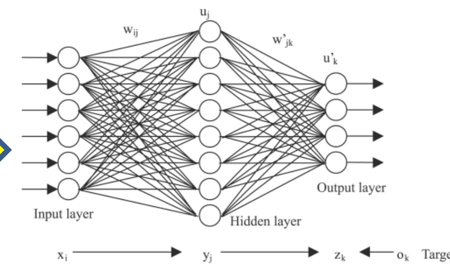


Even though XOR = (A & !B) || (B & !A), can't do it w/ 1 o/p

A single neuron is not enough

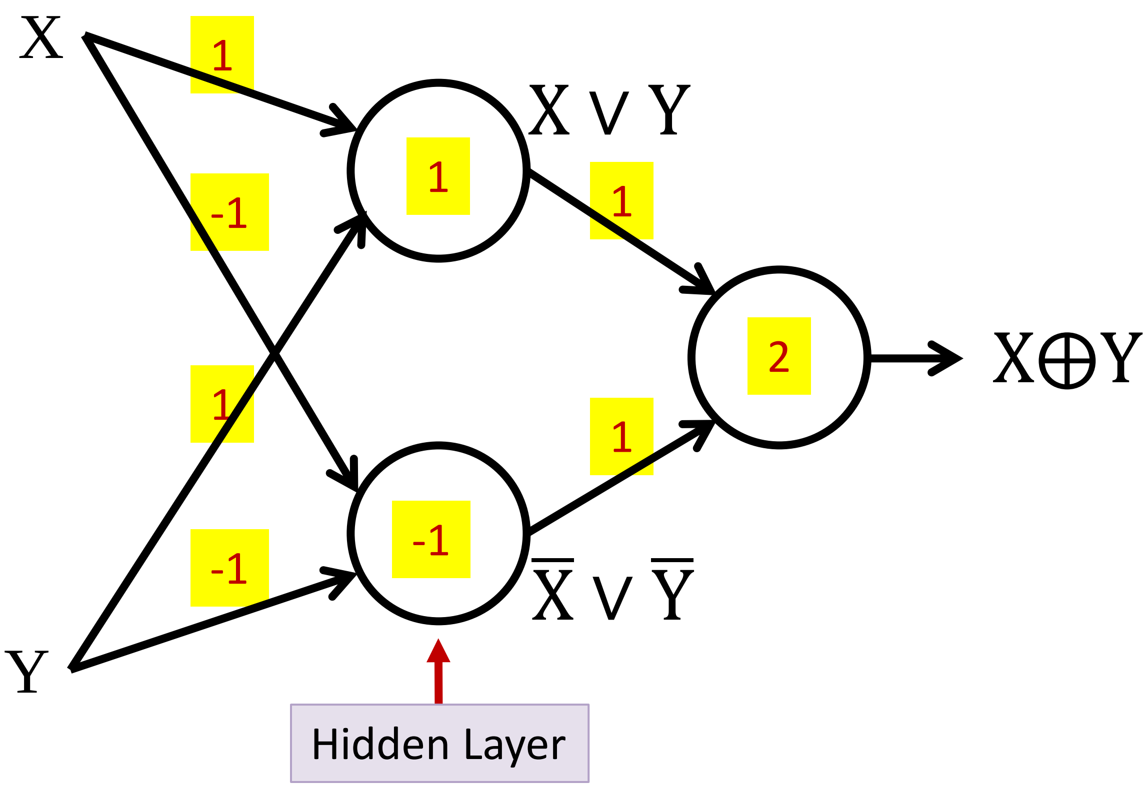






- Individual elements are weak computational elements
  - Marvin Minsky and Seymour Papert, 1969, *Perceptrons: An Introduction to Computational Geometry*
- *Networked* elements are required

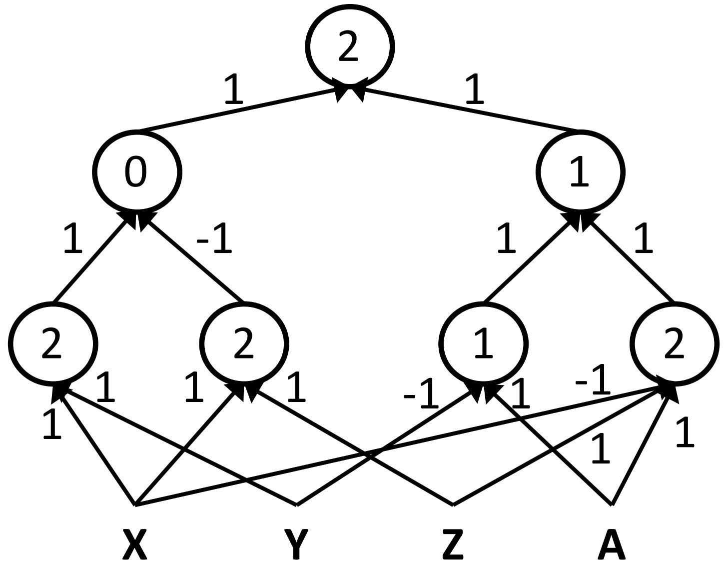
Multi-layer Perceptron!



- XOR
  - The first layer is a “hidden” layer

A more generic model

$$((A\&\bar{X}\&Z)|(A\&\bar{Y}))\&((X\ \&\ Y)|\overline{(X\&Z)})$$

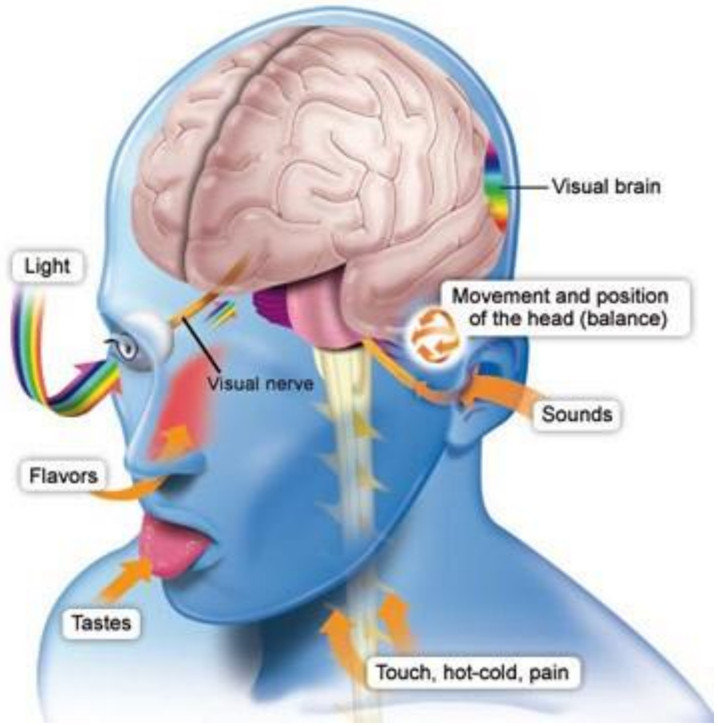


- A “multi-layer” perceptron
- Can compose arbitrarily complicated Boolean functions!
  - In cognitive terms: Can compute arbitrary Boolean functions over sensory input
  - More on this in the next class

Story so far

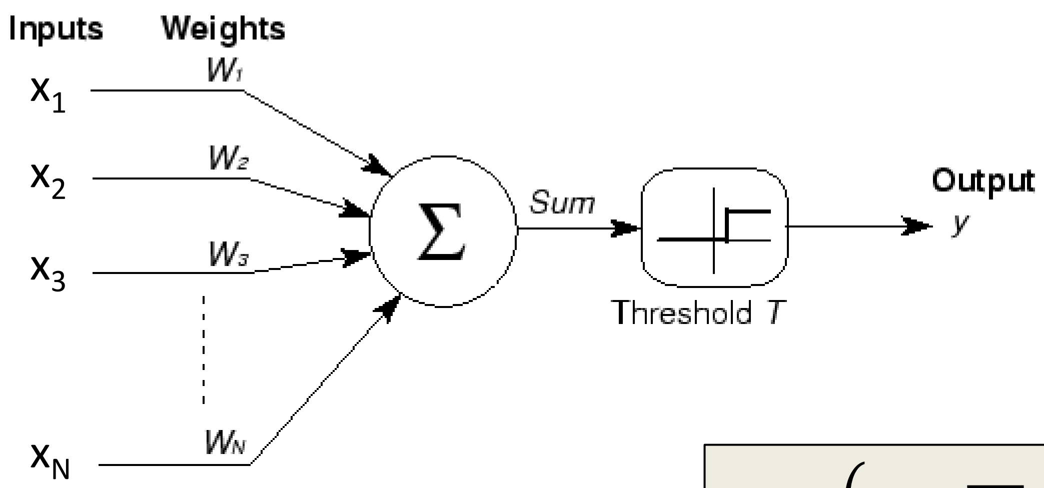
- Neural networks began as computational models of the brain
- Neural network models are *connectionist machines*
  – The comprise networks of neural units
- McCullough and Pitt model: Neurons as Boolean threshold units
  – Models the brain as performing propositional logic
  – But no learning rule
- Hebb's learning rule: Neurons that fire together wire together
  – Unstable
- Rosenblatt's perceptron : A variant of the McCulloch and Pitt neuron with a provably convergent learning rule
  – But individual units are limited in their capacity
- Multi-layer perceptrons can model arbitrarily complex Boolean functions

But our brain is not Boolean



- We have real inputs
- We make non-Boolean inferences/predictions

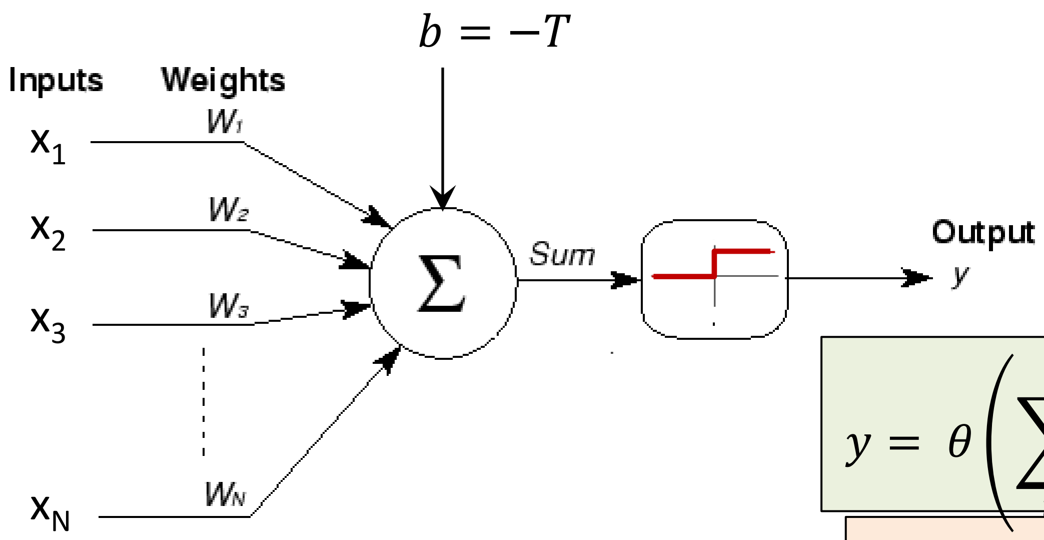
The perceptron with *real* inputs



$$y = \begin{cases} 1 \; if \; \sum_i w_i x_i - T \geq 0 \\ 0 \; else \end{cases}$$

- $x_1 \ldots x_N$ are real valued
- $w_1 \ldots w_N$ are real valued
- Unit "fires" if weighted input matches (or exceeds) a threshold

The perceptron with *real* inputs

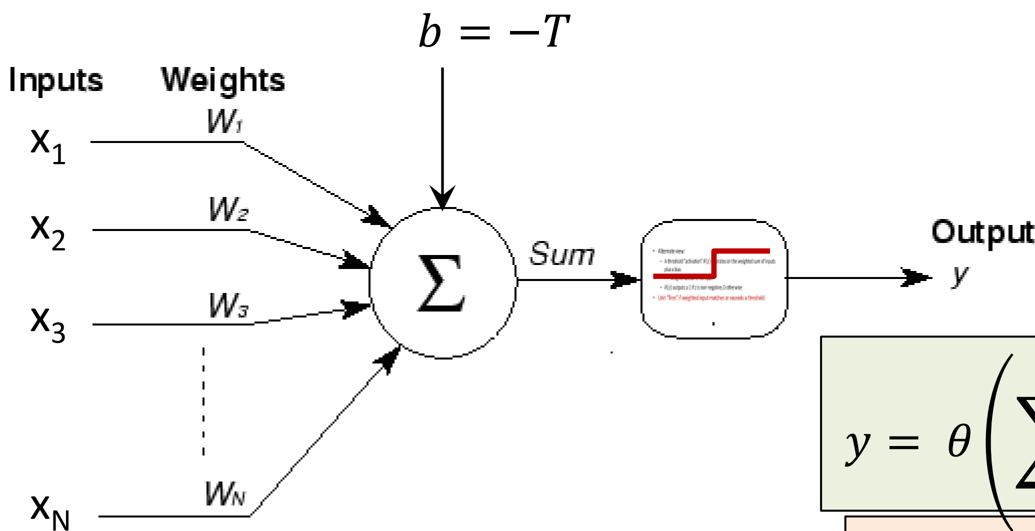


$$y = \theta\left(\sum_i w_i x_i + b\right)$$

$$\theta(z) = \begin{cases} 1 \; if \; z \geq 0 \\ 0 \; else \end{cases}$$

- Alternate view:
  - A threshold "activation" $\theta(z)$ operates on the weighted sum of inputs plus a bias
    - An *affine* function of the inputs affine = linear func + offset.
  - $\theta(z)$ outputs a 1 if z is non-negative, 0 otherwise
- Unit "fires" if weighted input matches or exceeds a threshold

The perceptron with *real* inputs



$$y = \theta\left(\sum_i w_i x_i + b\right)$$

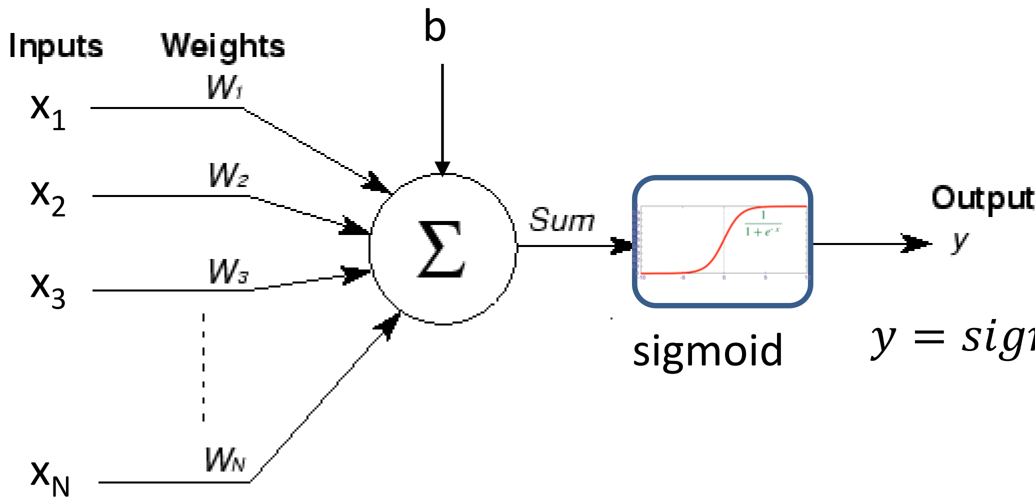
$$\theta(z) = \begin{cases} 1 \; if \; z \geq 0 \\ 0 \; else \end{cases}$$

- Alternate view:
  - A threshold "activation" $\theta(z)$ operates on the weighted sum of inputs plus a bias
    - An *affine* function of the inputs

What is the difference between "linear" and "affine"?

  - $\theta(z)$ outputs a 1 if z is non-negative, 0 otherwise
- Unit "fires" if weighted input matches or exceeds a threshold

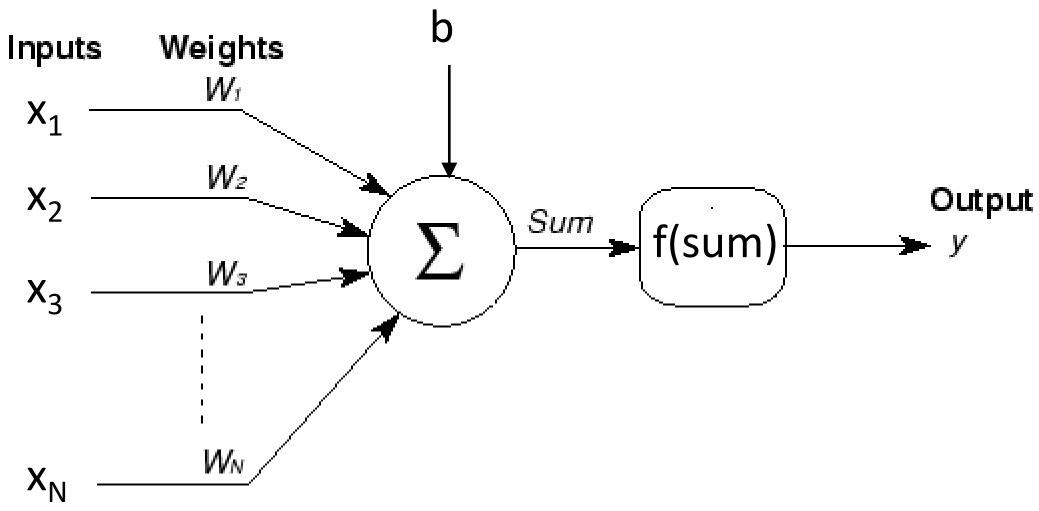
The perceptron with *real* inputs and a real *output*



$$y = sigmoid(\sum_i w_i x_i + b)$$

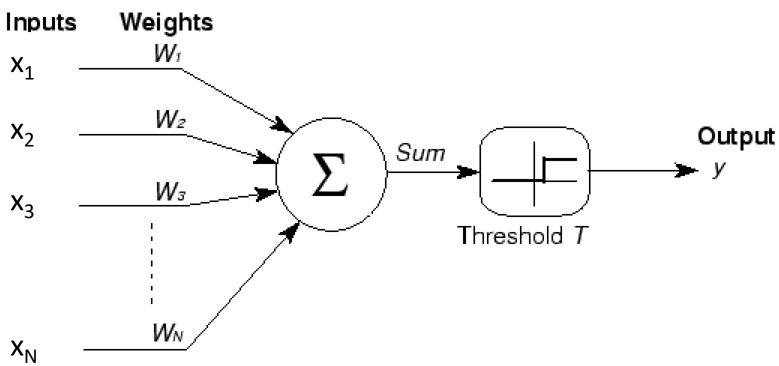
- $x_1 \ldots x_N$ are real valued
- $w_1 \ldots w_N$ are real valued
- The output y can also be real valued
  - Sometimes viewed as the "probability" of firing

The "real" valued perceptron

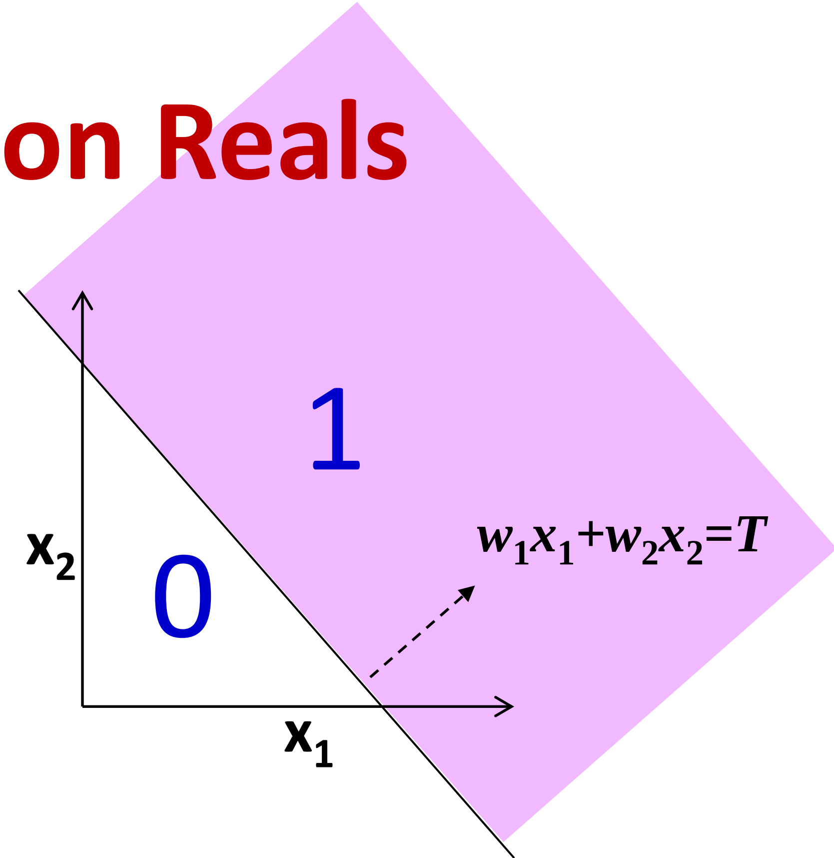


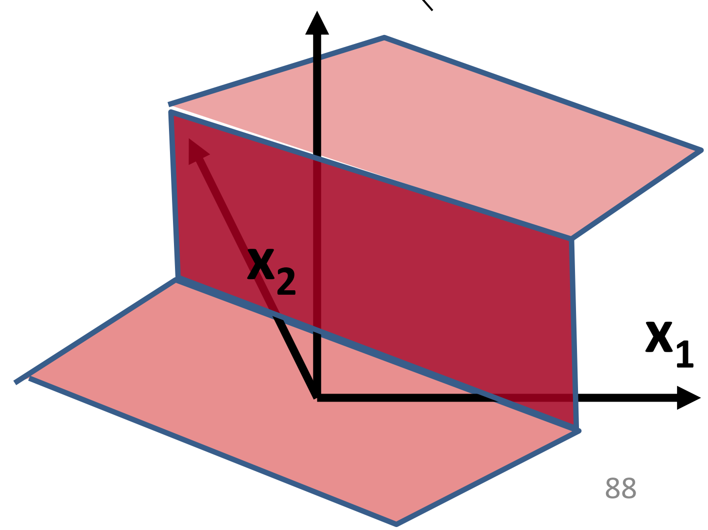
- Any real-valued "activation" function may operate on the affine function of the input
  - We will see several later
  - Output will be real valued
- The perceptron maps real-valued inputs to real-valued outputs
- *Is useful to continue assuming Boolean outputs though, for interpretation*

A Perceptron on Reals



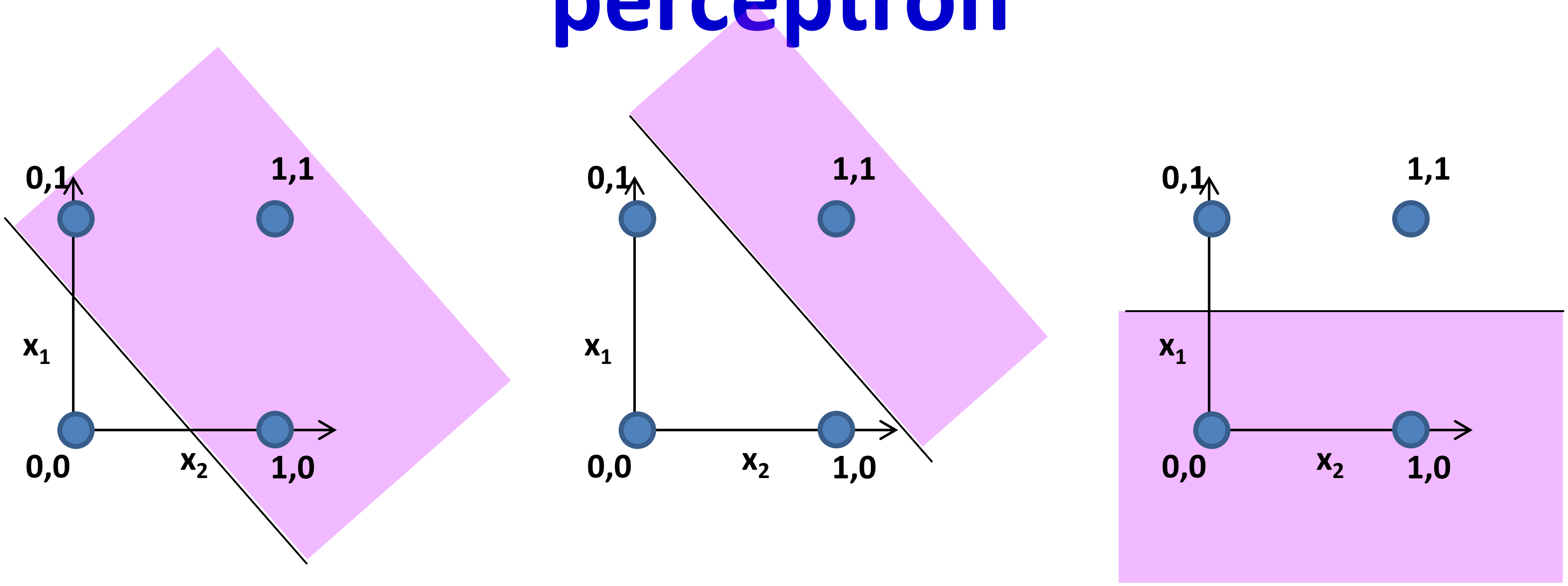
$$y = \begin{cases} 1 \; if \; \sum_i w_i \mathrm{x}_i \geq T \\ 0 \; else \end{cases}$$





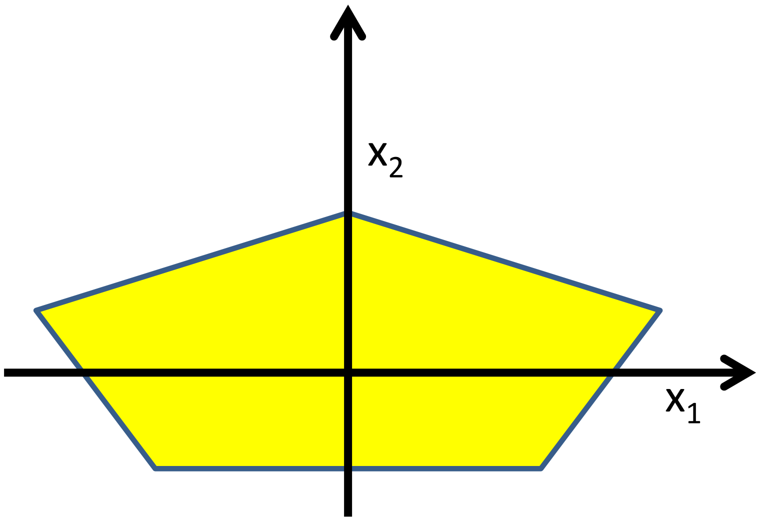
- A perceptron operates on *real*-valued vectors
  - **This is a *linear classifier***

Boolean functions with a real perceptron



- Boolean perceptrons are also linear classifiers
  - Purple regions have output 1 in the figures
  - What are these functions
  - Why can we not compose an XOR?

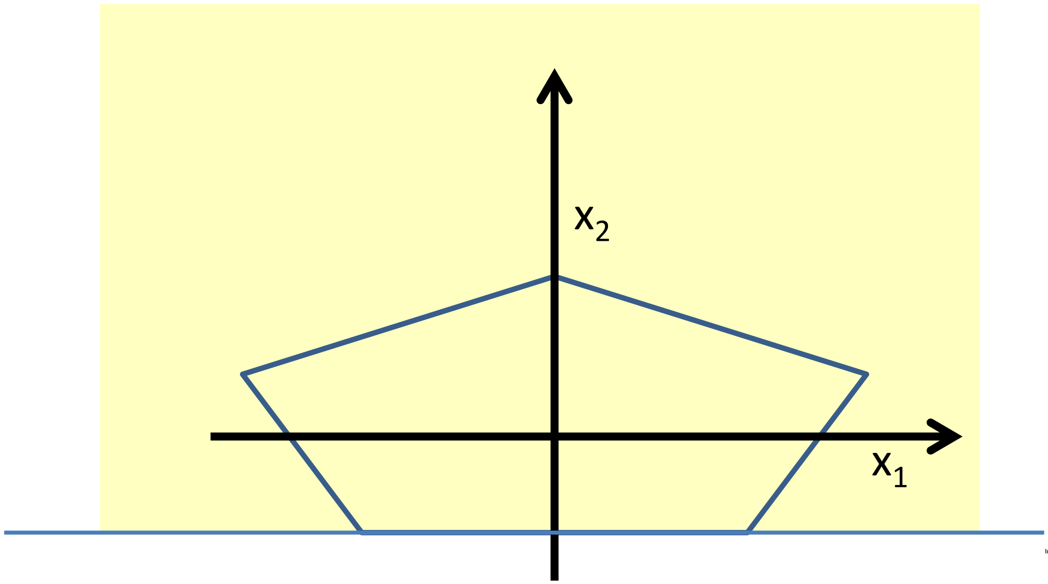
Composing complicated "decision" boundaries

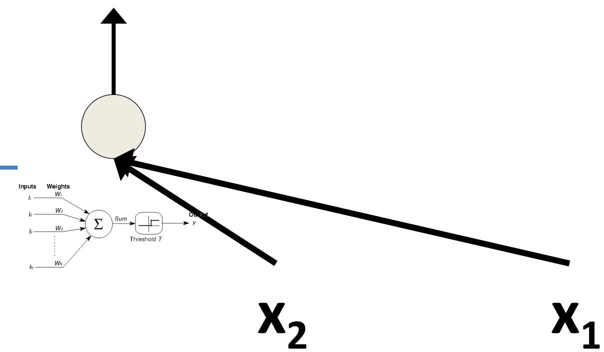


Can now be composed into "networks" to compute arbitrary classification "boundaries"

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

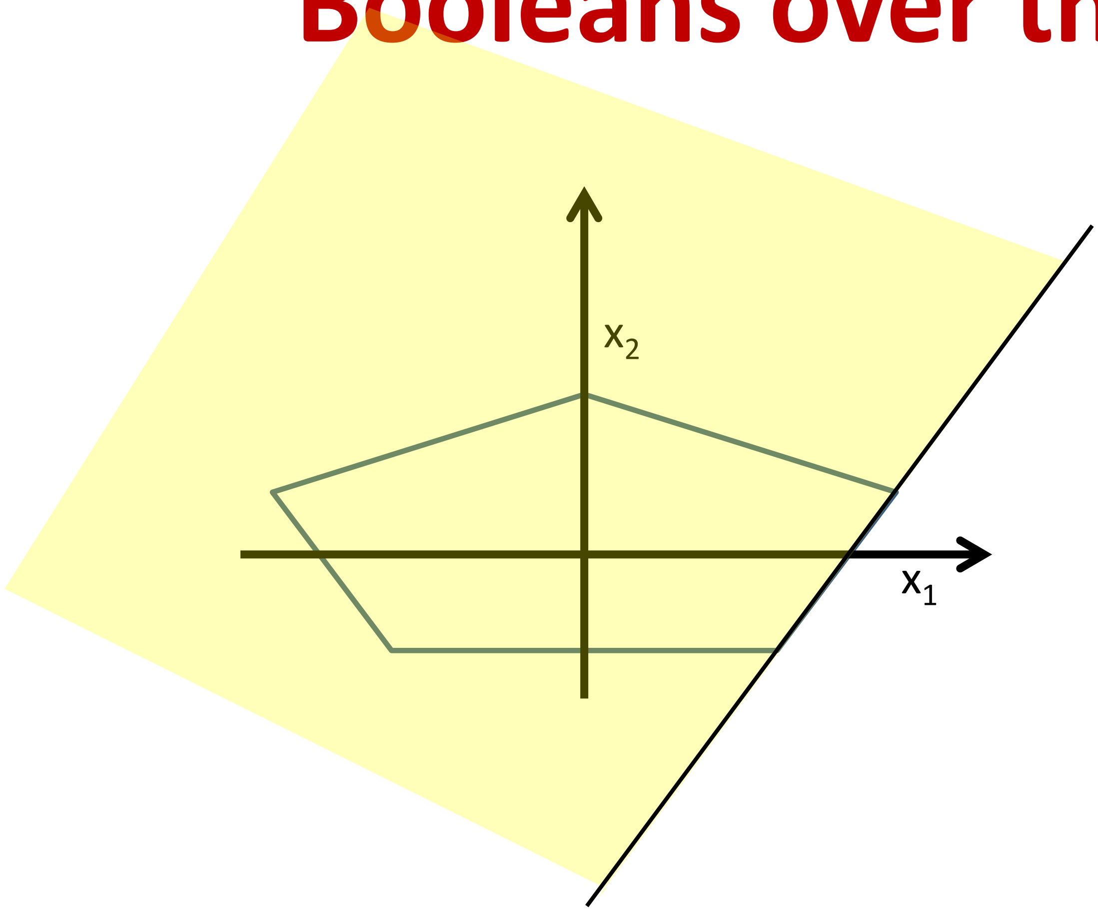
Booleans over the reals

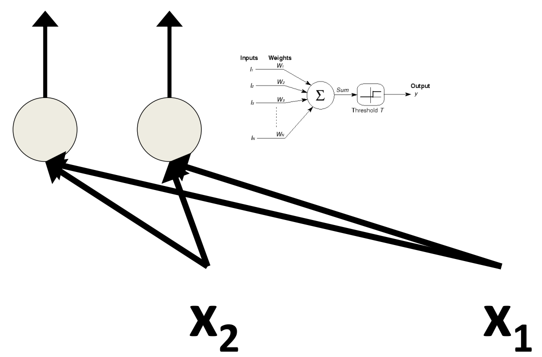




- The network must fire if the input is in the coloured area

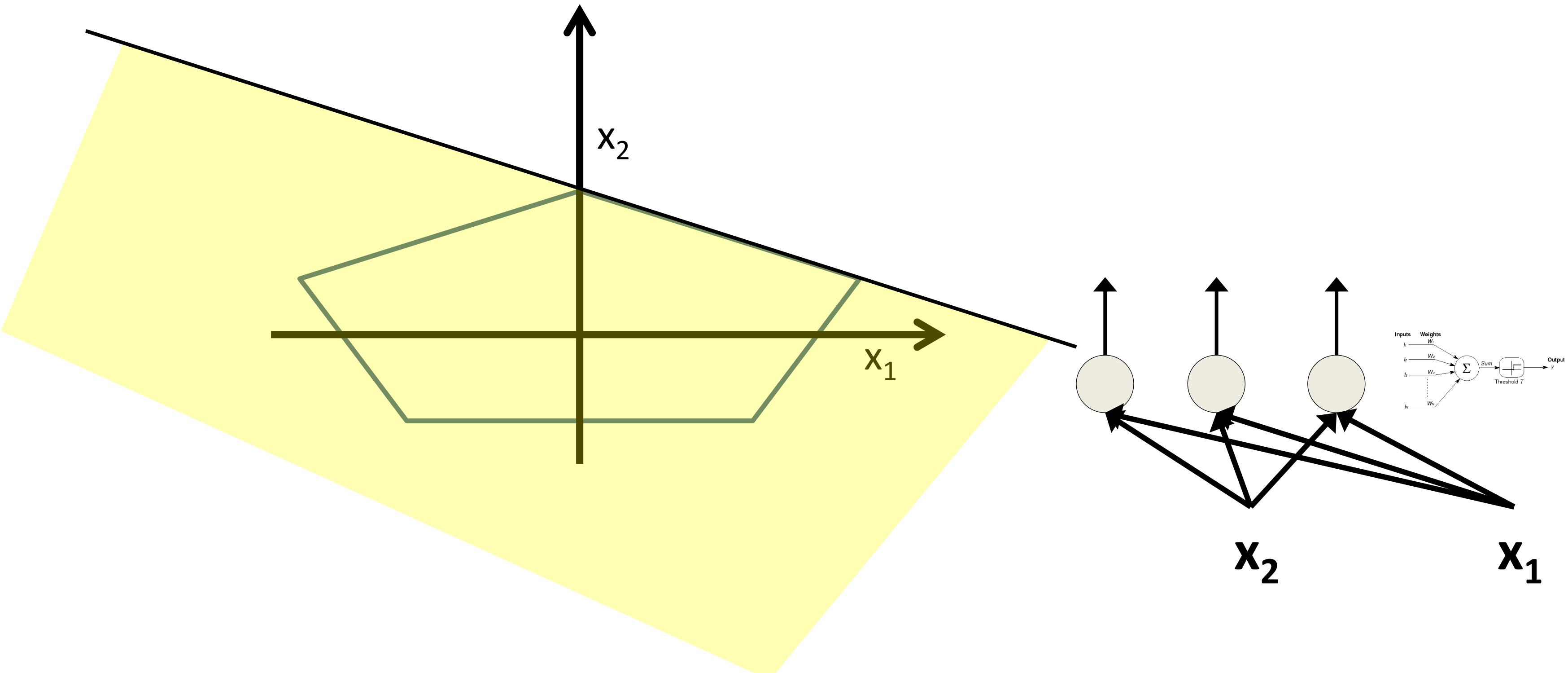
Booleans over the reals





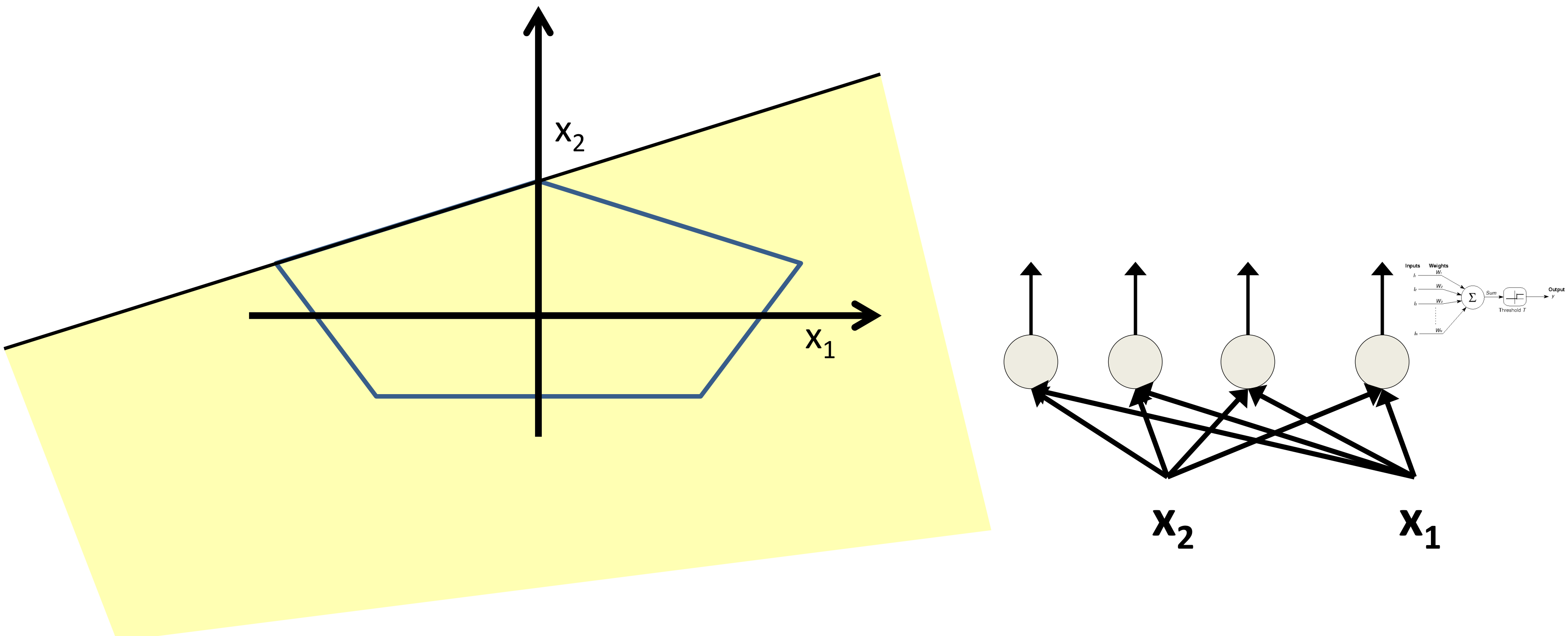
- The network must fire if the input is in the coloured area

Booleans over the reals



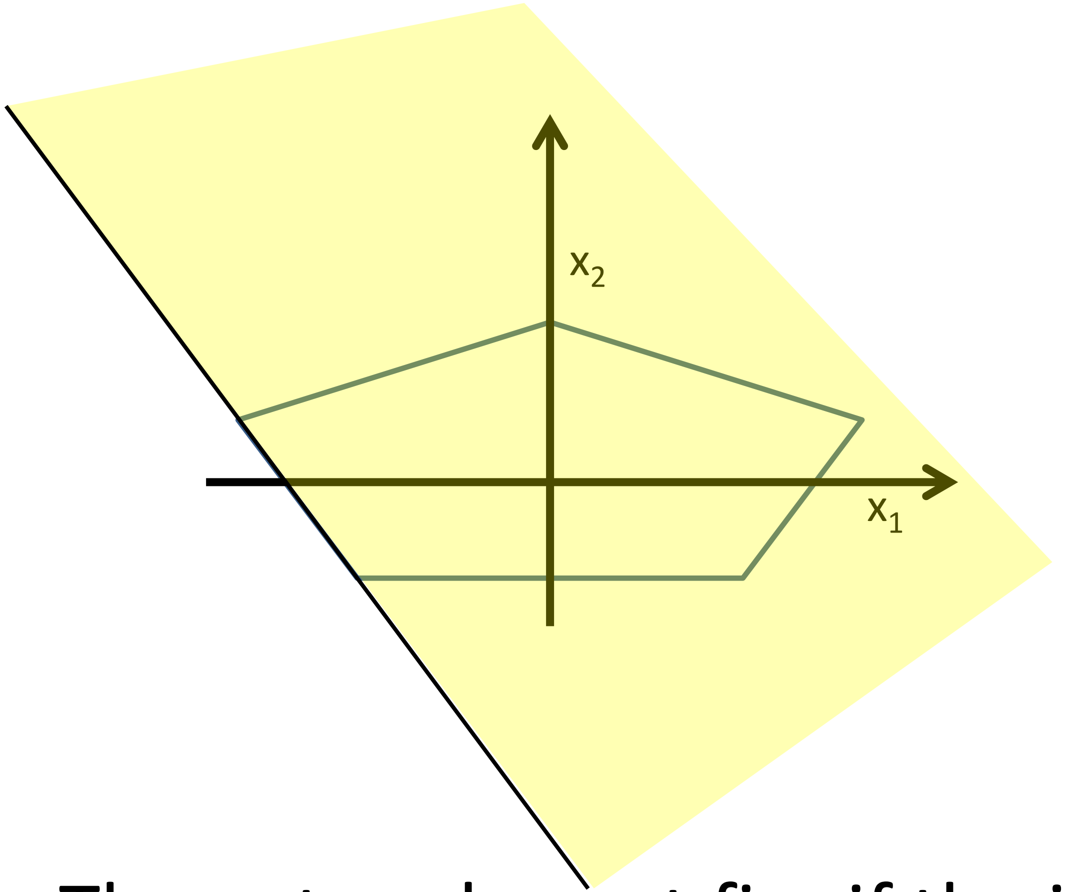
- The network must fire if the input is in the coloured area

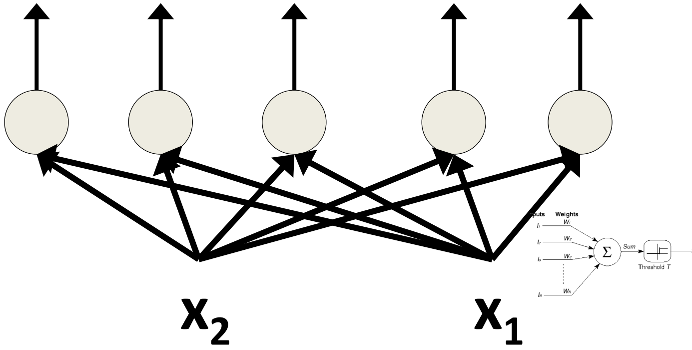
Booleans over the reals



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

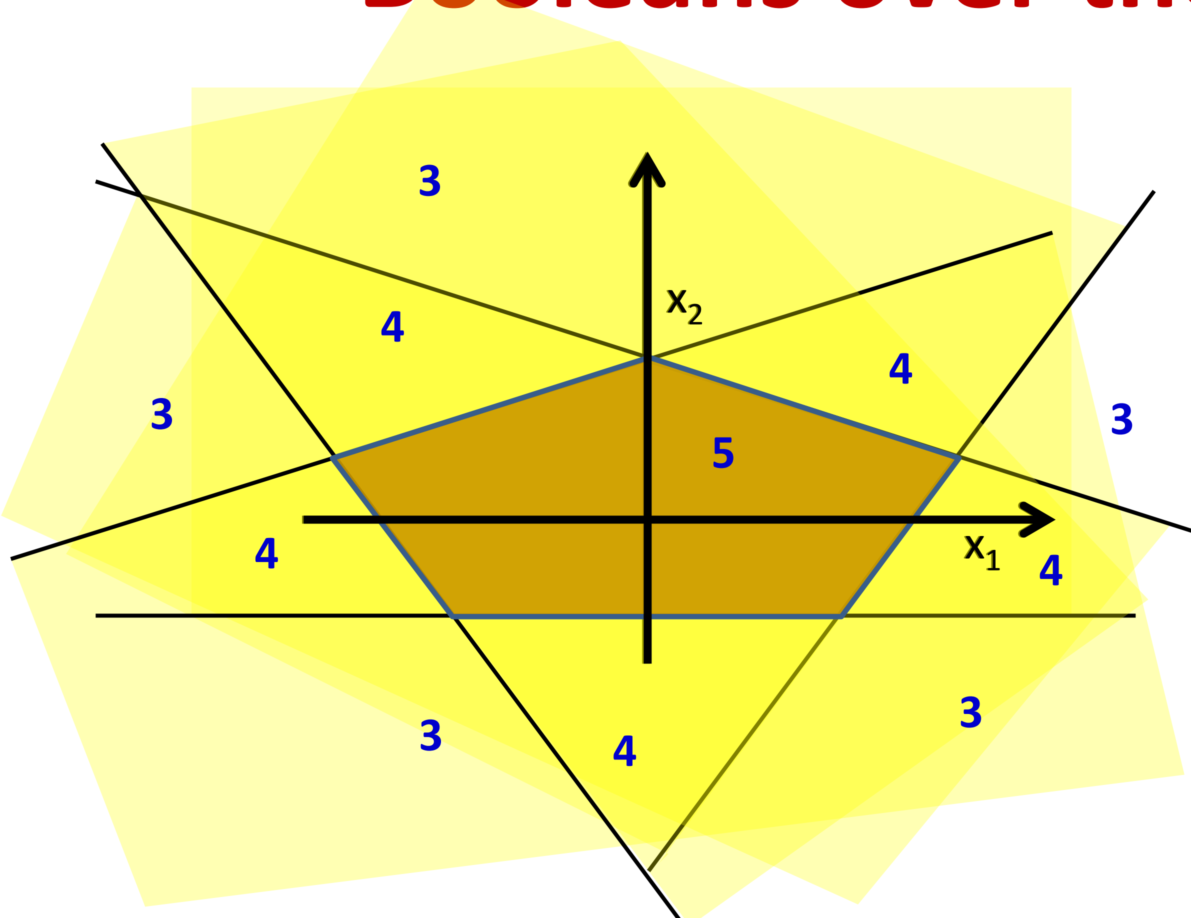
Booleans over the reals

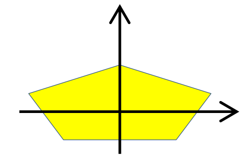




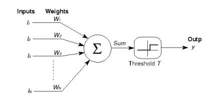
- The network must fire if the input is in the coloured area

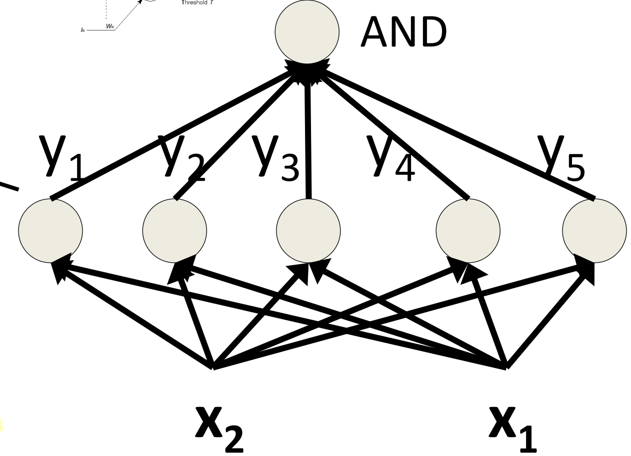
Booleans over the reals





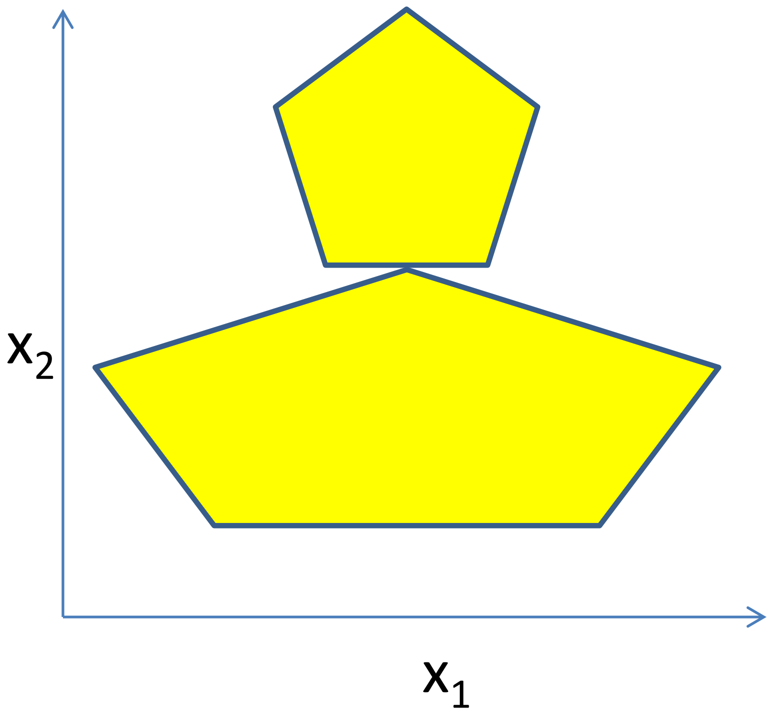
$$\sum_{i=1}^{N} y_i \geq 5?$$

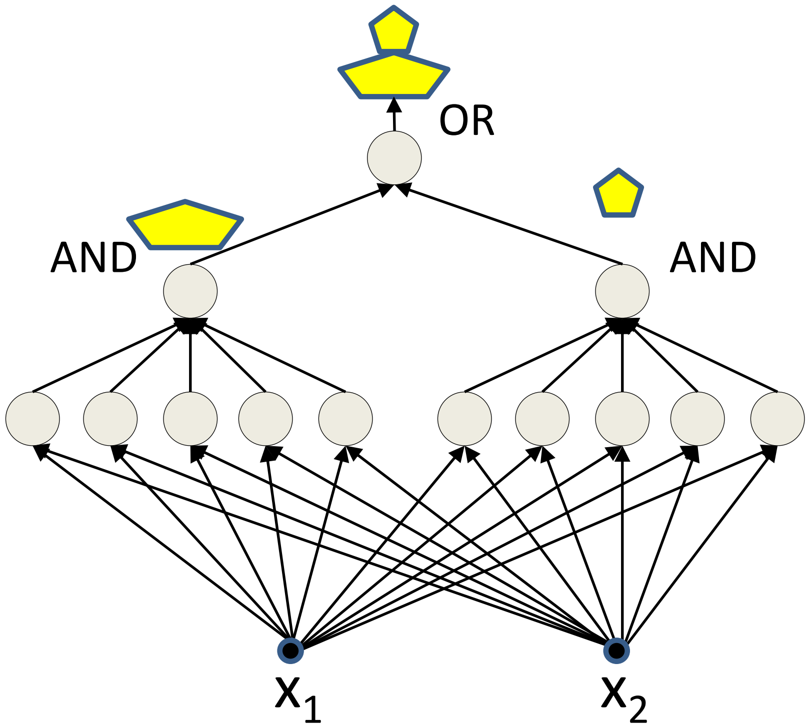




- The network must fire if the input is in the coloured area

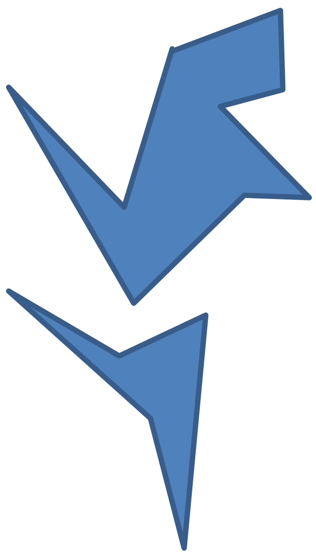
More complex decision boundaries

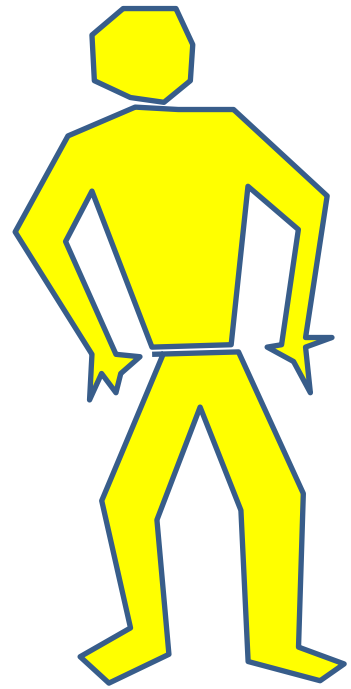


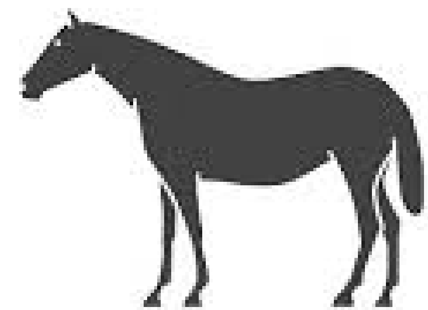


- Network to fire if the input is in the yellow area
  - "OR" two polygons
  - A third layer is required

Complex decision boundaries



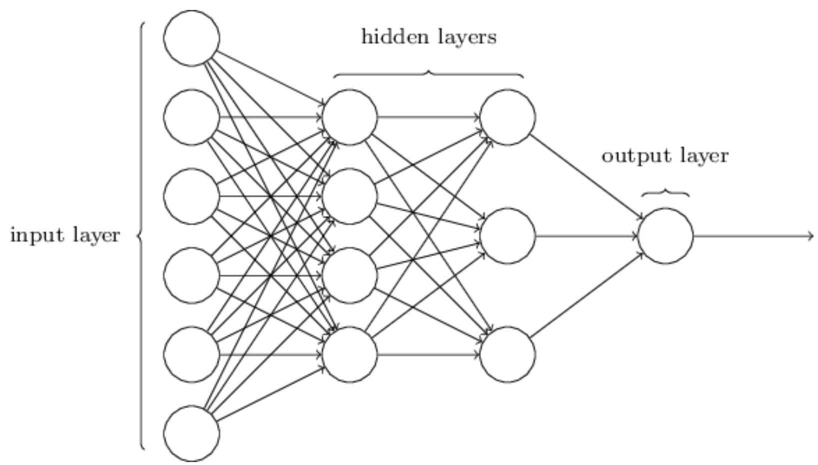


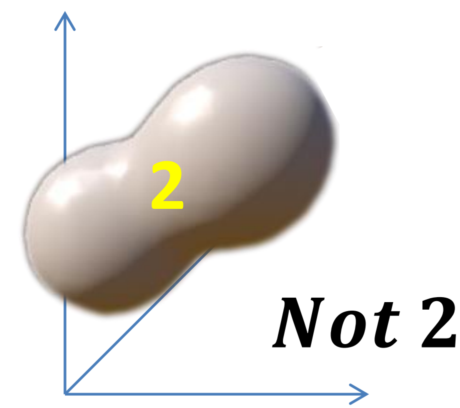


- Can compose very complex decision boundaries
  - How complex exactly? More on this in the next class

Complex decision boundaries

784 dimensions
(MNIST)





784 dimensions

- Classification problems: finding decision boundaries in high-dimensional space
  - Can be performed by an MLP
- MLPs can *classify* real-valued inputs

Story so far

- **MLPs are connectionist computational models**
  - Individual perceptrons are computational equivalent of neurons
  - The MLP is a layered composition of many perceptrons
- **MLPs can model Boolean functions**
  - Individual perceptrons can act as Boolean gates
  - Networks of perceptrons are Boolean functions
- **MLPs are Boolean *machines***
  - They represent Boolean functions over linear boundaries
  - They can represent arbitrary decision boundaries
  - They can be used to *classify* data

Poll 3

How many threshold activation perceptrons will we need in an MLP to model a hexagonal decision region (a decision region bounded by a six-sided polygon) over a two-dimensional input space? (Single Choice)

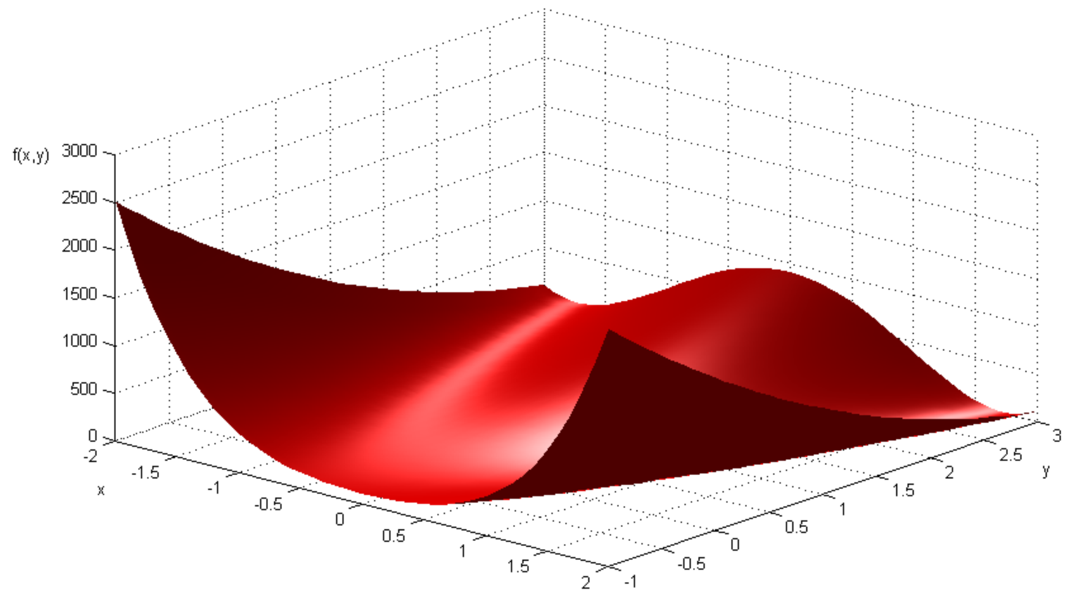
- 6
- 7
- 12
- 13

Poll 3

How many threshold activation perceptrons will we need in an MLP to model a hexagonal decision region (a decision region bounded by a six-sided polygon) over a two-dimensional input space? (Single Choice)

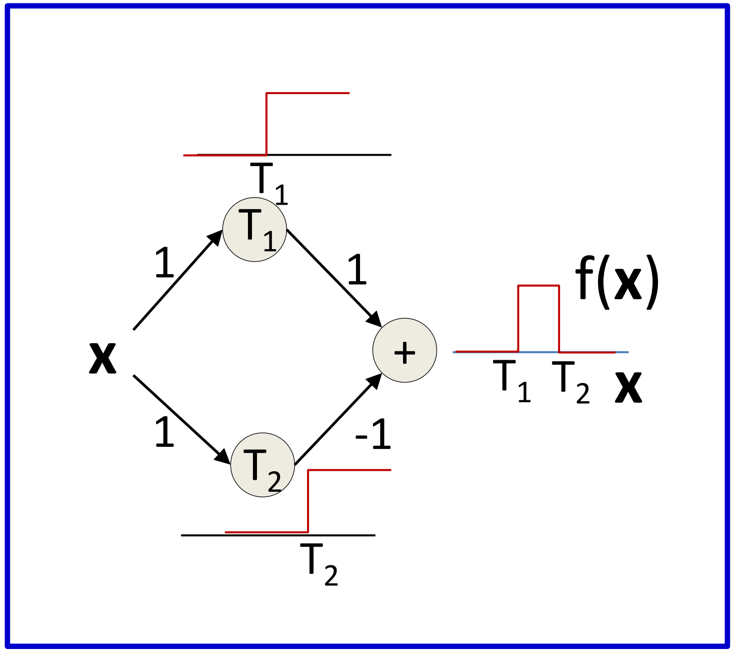
- 6
- **7**
- 12
- 13

But what about continuous valued *outputs?*



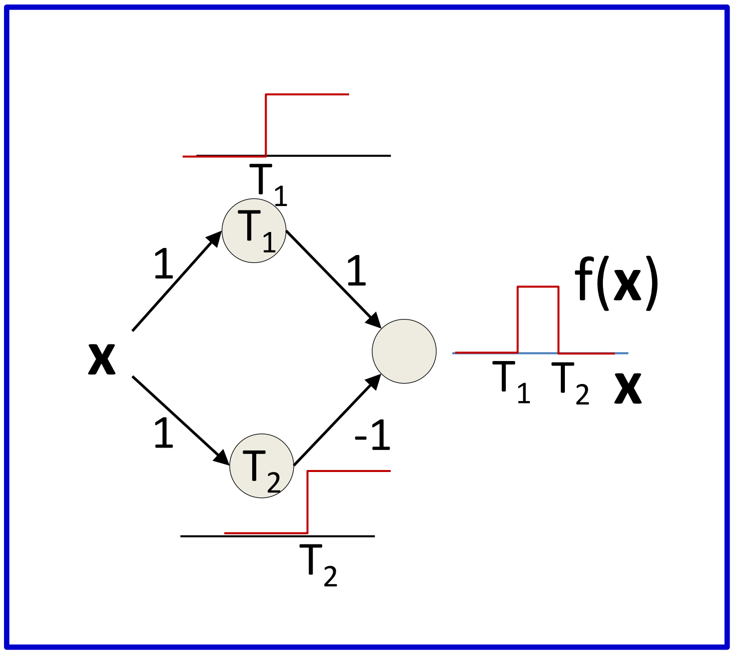
- Inputs may be real-valued
- Can outputs be continuous-valued too?

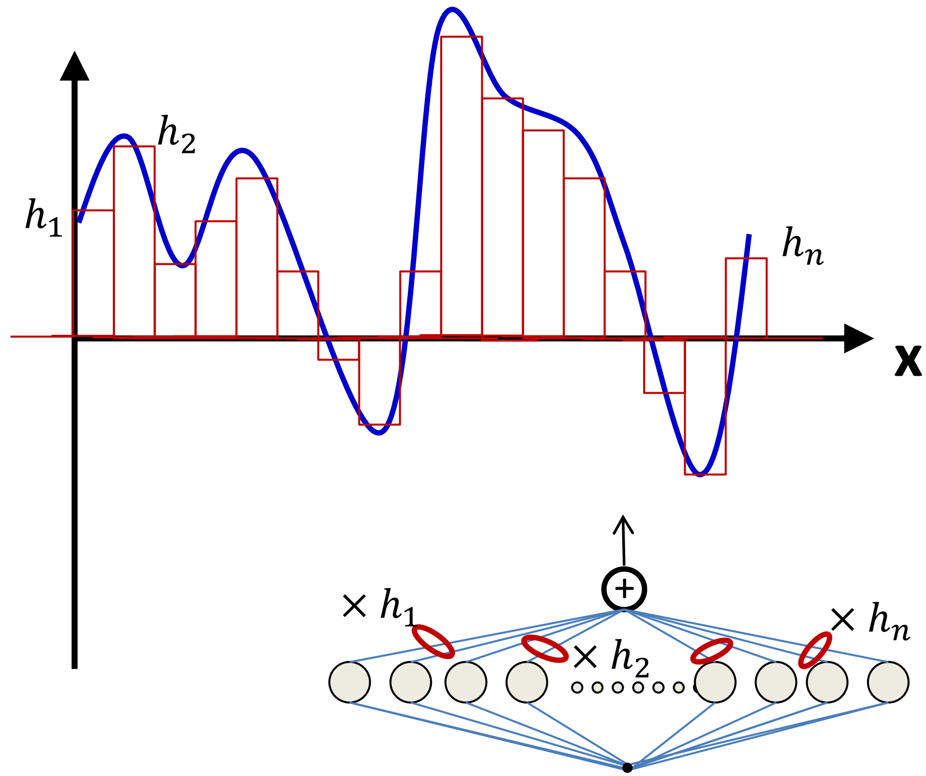
MLP as a continuous-valued regression



- 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
- This generalizes to functions of any number of inputs (next class)

Poll 4

How many neurons will be required by a network of sinusoidal (y = sin(z)) activation neurons to precisely model the scalar function y = cos(2x) (for scalar input x)? (Single Choice)

- 3
- floor(pi/2) or ceil(pi/2)
- infinite
- none of the above

Poll 4

How many neurons will be required by a network of sinusoidal (y = sin(z)) activation neurons to precisely model the scalar function y = cos(2x) (for scalar input x)? (Single Choice)

- 3
- floor(pi/2) or ceil(pi/2)
- infinite
- **none of the above**

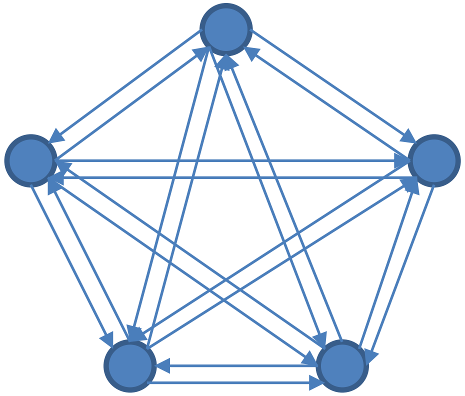
Explanation: you only need one

Story so far

- **Multi-layer perceptrons are connectionist computational models**
- **MLPs are *classification engines***
  - They can identify classes in the data
  - Individual perceptrons are feature detectors
  - The network will fire if the combination of the detected basic features matches an "acceptable" pattern for a desired class of signal
- **MLP can also model continuous valued functions**

Other things MLPs can do

- Model memory
  - Loopy networks can "remember" patterns
    - Proposed by Lawrence Kubie in 1930, as a model for memory in the CNS



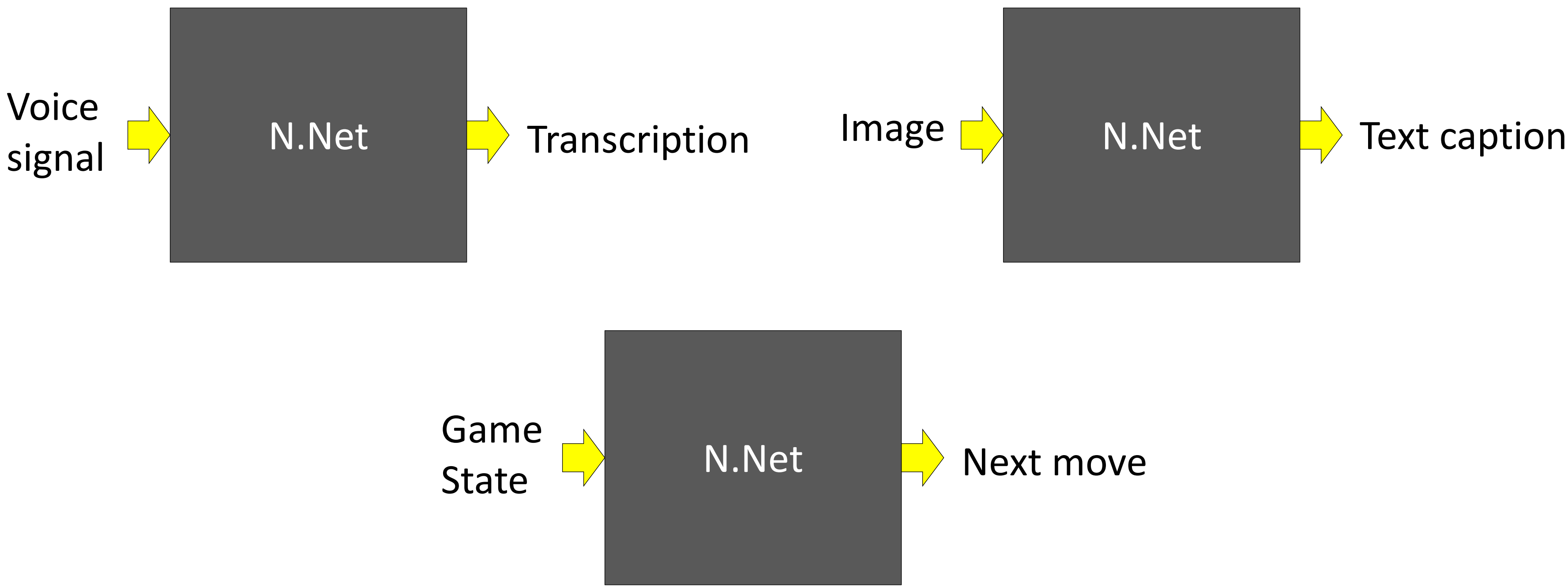
- Represent probability distributions
  - Over integer, real and complex-valued domains
  - MLPs can model both *a posteriori* and *a priori* distributions of data
    - A posteriori conditioned on other variables
  - MLPs can *generate* data from complicated, or even unknown distributions

- They can rub their stomachs and pat their heads at the same time..

NNets in AI

- The network is a function
  - Given an input, it computes the function layer wise to predict an output
    - More generally, given one or more inputs, predicts one or more outputs

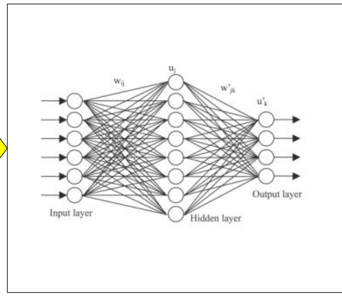
These tasks are *functions*



- Each of these boxes is actually a function
  - E.g f: Image → Caption

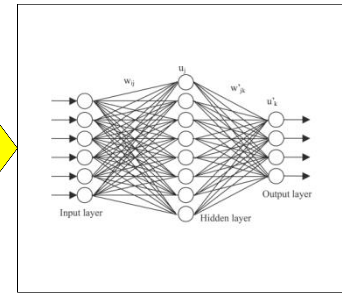
These tasks are *functions*

Voice signal



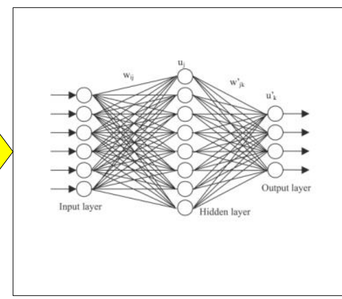
Transcription

Image



Text caption

Game State



Next move

- Each box is actually a function
  - E.g f: Image → Caption
  - It can be approximated by a neural network

Story so far

- Multi-layer perceptrons are connectionist computational models
- MLPs are *classification engines*

- MLP can also model continuous valued functions
- Interesting AI tasks are functions that can be modelled by the network

Today's lessons

- A brief history of neural networks
  - Connectionism
    - Its relation to cognition and the brain
    - Its contrast to conventional computer architecture
  - Early models, and their limitations
- Introducing modern neural networks
- And what they can compute

Next Up

- More on neural networks as universal approximators
  - And the issue of depth in networks