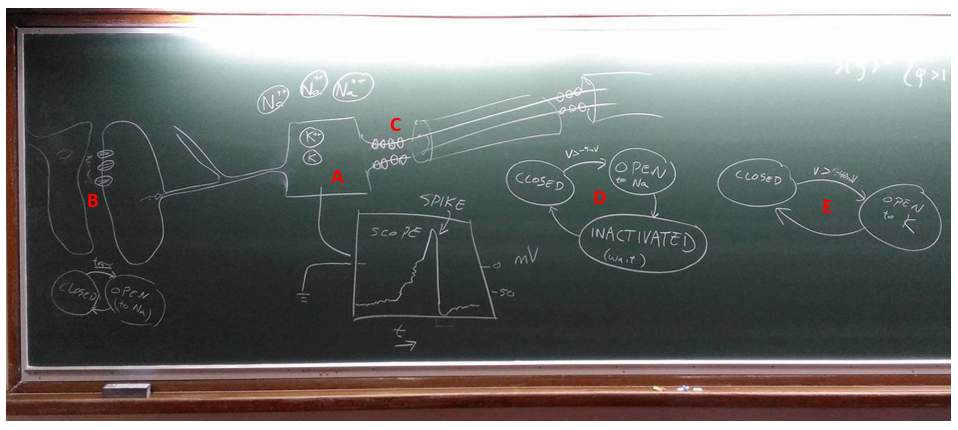
Lecture 3

Scribe – Deirdre Cleary

Neurons are responsible for performing actions in the brain.

They account for <10% of cells in the brain.

The rest of the cells are called glia.



Please note that I added the labels for convenience to add extra notes and that they were not part of Barak’s diagram.

A – A reservoir of salt with some voltage difference between the inside and the outside (≈65 to 70 mV).

B – A gate (blown up in size) which lets Na+ ions in. The state diagram below shows that the gate changes from closed to open as a result of some transmitter.

C – The area marked by C is the spike initiation zone. Here there are a series of voltage sensitive ion channels which depend on voltage in the membrane. Along the axon there are a number of myelin sheaths followed by ion channels which propagate the spike along.

D – A state diagram which shows the transition of an ion channel from closed to open to K (when V > -40mV) and back.

E – A state diagram which shows the transition of an ion channel from closed, to open to Na (when V > 5mV) which sends V way up, to inactivated for some time.

Neurons expend a lot of energy. They expend K ions when there are more inside than out, or take in Na ions when there are more outside than in.

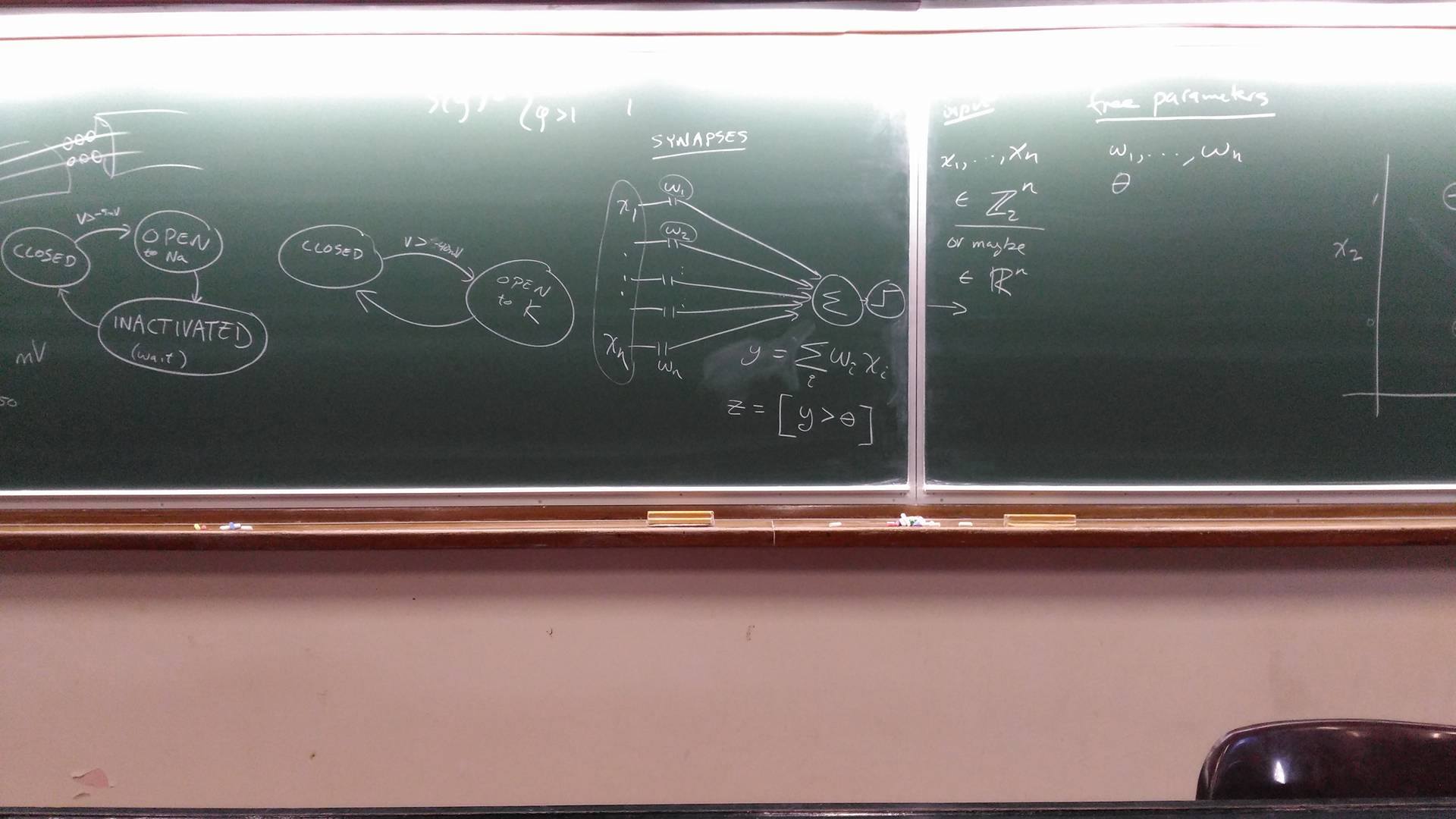
# Abstracting to a computer

McCulloch and Pitts developed a neuron model in 1943.

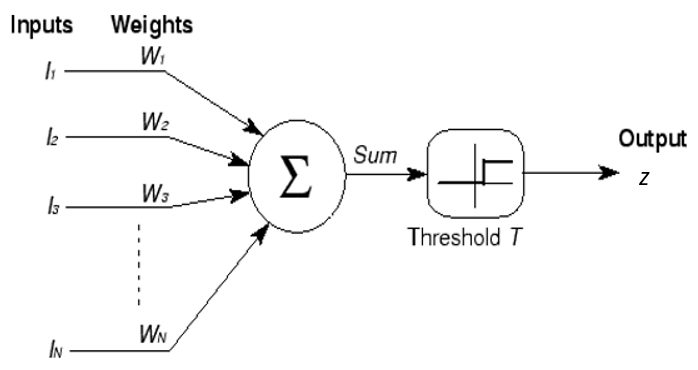
They suggested that voltage is a linear sum of charges applied to synapses.

At the time the delays that occur in the brain did not seem unreasonable to model as computers at the time were quite slow themselves.

# Synapses



Or for a clearer image,



Here the variable strength (w) is how much charge is injected.

We can use a number of representations for the output, z.



where

* Knuth notation
* +/- 1 depending on the Boolean in []

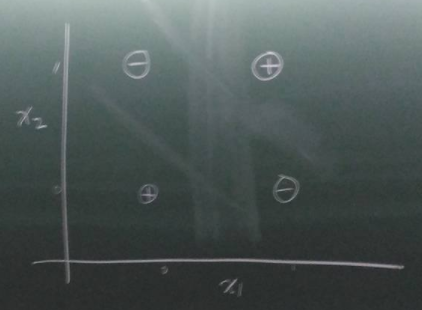
Note that instead of 0 we can use some threshold θ.

Free parameters – w1, … , wn and θ.

Input – x1, … , xn ϵ or maybe ϵ

* Outputs of other neurons.

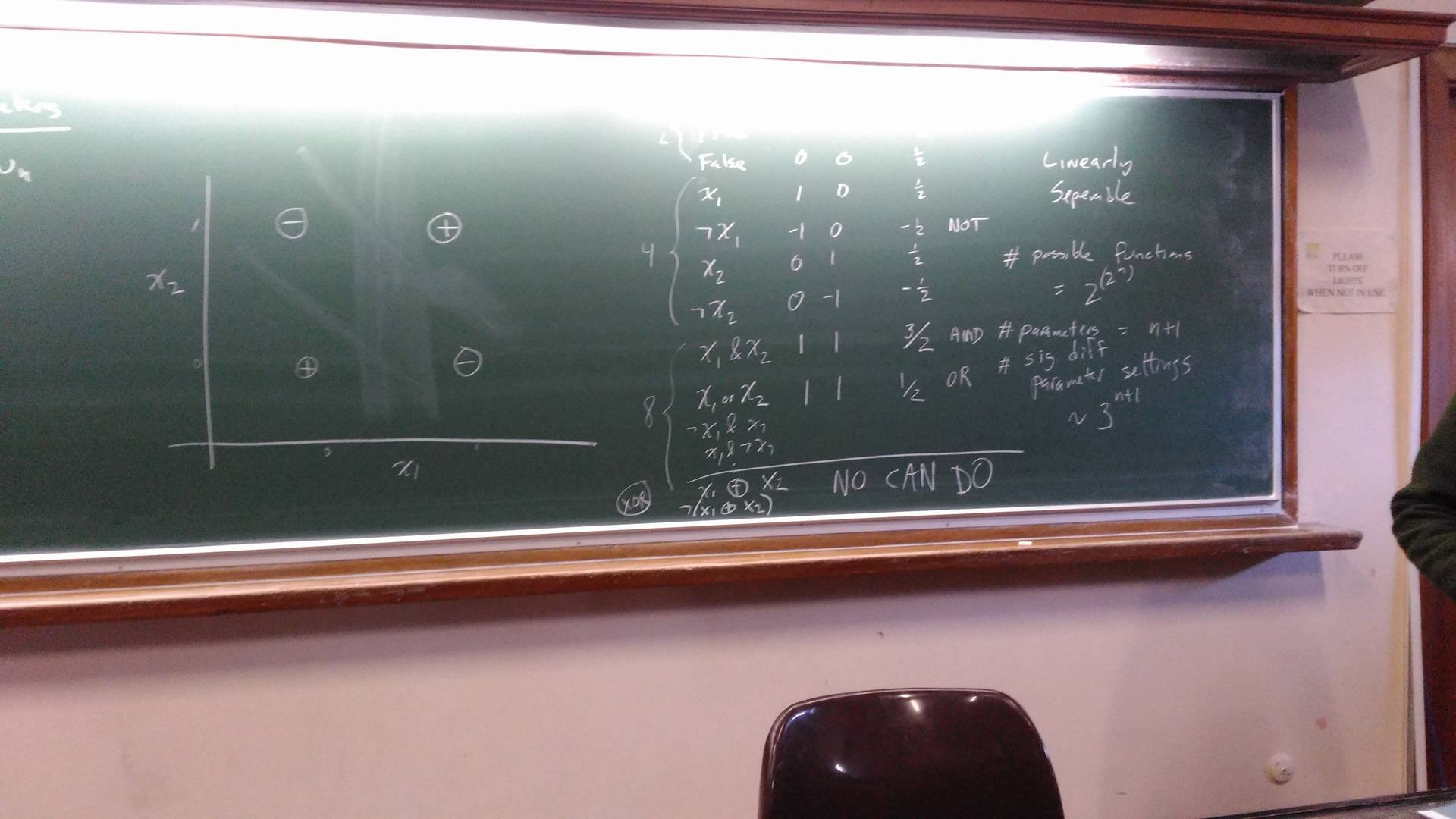
Inputs to data set live in input space.



(Excuse the blurry image. Signs shown are for XOR case)

Note that the graph above is in 2D rather than n-dimensions as it was being drawn on the blackboard.

What Boolean functions can it compute?



Or without the glare,

|  |  |  |  |
| --- | --- | --- | --- |
| Expression | w1 | w2 | Θ |
| True | 0 | 0 | -½ |
| False | 0 | 0 | ½ |
| x1 | 1 | 0 | ½ |
| ¬x1 | -1 | 0 | -½ |
| x2 | 0 | 1 | ½ |
| ¬x2 | 0 | -1 | -½ |
| x1 &x2 | 1 | 1 | 3/2 |
| x1 ORx2 | 1 | 1 | ½ |
| ¬x1 &x2 |  |  |  |
| x1 &¬x2 |  |  |  |
| … |  |  |  |
| x1 XORx2 | NO | CAN | DO |
| ¬(x1 XORx2) | NO | CAN | DO |

The threshold states how much “oomph” it needs.

The XOR case above is difficult as no one line can be drawn to separate the positive and negative spaces. Thus, it is not linearly separate.

So, we can implement 14/16 expressions.

As n gets high, the number of functions that are linearly separate gets small.

# possible functions =

# parameters = n + 1

# significantly different parameter settings ~ 3n+1

We want to be in a space where knowing values on some inputs generates those on others.

McCulloch and Pitts began making circuits of them.

They formed a superset of conventional logic.

Eg. output is true if majority are on – difficult to do in standard circuit.

Flies have short lifespans. They are thought to be “wired up” at birth (debateable). Some animals, by contrast, have to do a lot of learning eg. humans.

Rosenblatt discovered that ws are easy to modify in biology.

Idea (continued in next lecture)

* w = real number
* output was strong enough or not
* didn’t fire but should have
* if 0, didn’t enter computation
* make wi bigger if should have fired but didn’t or smaller if fired but shouldn’t have