Let's look at the problem of recognising hand-written digits.
Good toy problem: practical, hard, simple.
It will take us the whole lecture to do a first draft on how to solve this problem.

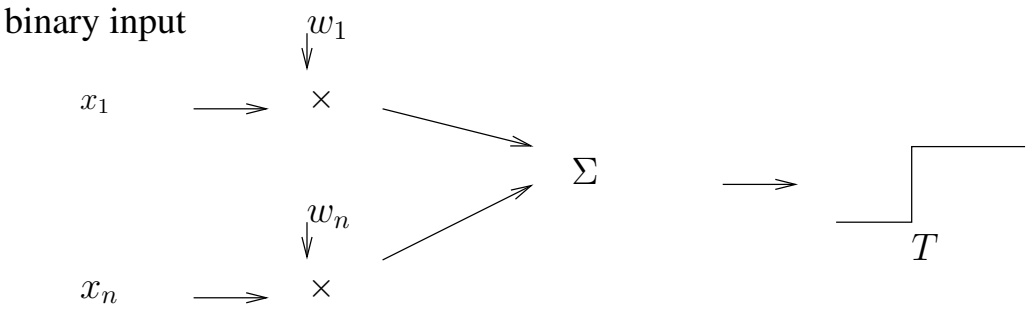
Humans can do it, so maybe think about how we might do it.

Neural Networks

ANN

- Long a curiosity
- 2012 paper, Hinton, image classification, 1000 categories, 60 million parameters
- neurones : axone, terminaison de l'axone, noyau, dendrites; activation energy

Show a neuron with dendrites, explain how it works.



What are we modeling?

1. all or none
2. cumulative influence
3. synaptic weight
4. (not) refractory period *(période réfractaire)*
5. (not) axonal bifurcation
6. (not) time patterns

So we have a model of a neuron (a collection of weights and thresholds).
But what about collections of neurons?

$$\text{inputs} \rightarrow (w, t) \rightarrow \text{outputs}$$

So we want $z = f(x, w, t)$. Training means adjusting w, t.
An ANN is a function approximator.

We want some desired function, $d = g(x)$.

Types of neurons

Linear neuron

$$y = b + \sum_i x_i w_i$$

where

$$\begin{aligned} y &= \text{output} \\ b &= \text{bias} \\ x_i &= i^{\text{th}} \text{ input} \\ w_i &= \text{weight on } i^{\text{th}} \text{ input} \end{aligned}$$

Binary threshold neuron

$$z = \sum_i x_i w_i$$

$$y = \begin{cases} 1 \text{ if } z \geqslant 0 \\ 0 \text{ otherwise} \end{cases}$$

where

$$\begin{aligned} z &= \text{total input} \\ y &= \text{output} \\ x_i &= i^{\text{th}} \text{ input} \\ w_i &= \text{weight on } i^{\text{th}} \text{ input} \end{aligned}$$

Rectified linear neuron

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} z \text{ if } z \geqslant 0 \\ 0 \text{ otherwise} \end{cases}$$

where

$$
\begin{aligned}
z &= \text{total input} \\
y &= \text{output} \\
b &= \text{bias} \\
x_i &= i^{\text{th}} \text{ input} \\
w_i &= \text{weight on } i^{\text{th}} \text{ input}
\end{aligned}
$$

Sigmoid neuron

$$z = b + \sum_i x_i w_i$$

$$y = \frac{1}{1 + e^{-z}}$$

(This is differentiable.)

Stochastic binary neuron

$$z = b + \sum_i x_i w_i$$

$$p = \frac{1}{1 + e^{-z}}$$

$$y = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ with probability } 1 - p \end{cases}$$

(a probability distribution)

Can also do something similar with rectified linear neurons, produce spikes with probability p with a Poisson distribution.

Neural Networks

Architecture is how we connect the states.

Feed forward network

- Flow is unidirectional
- No loops