Afstudeerborrel, Hofwijck, NL

Imperial College London

Deduction & Induction
Two ingredients for building machine intelligence.

Mark van der Wilk

Department of Computing
Imperial College London
https://mvdw.uk

@markvanderwilk
m.vdwilk@imperial.ac.uk

Oct 22, 2022

Why did I invite you here?

I thought it was worth celebrating that I graduated from my PhD

Dutch-style graduations are a bigger get-together though.

Why did I invite you here?

I thought it was worth celebrating that I graduated from my PhD
... in 2019.

Dutch-style graduations are a bigger get-together though.

Today

- What led me to the questions that I'm researching.
- Some key ideas that underpin my work.
- The Big Questions.

What I do these days

- I work on the mathematics of **learning**
- To build systems that learn by experience
  to reach useful goals
- I mainly work on theory, but connects to lots of applications
  - Getting computers to understand images
  - Minimising waste in chemical processes (BASF)
  - Increasing efficiency of aeroplane engines (Secondmind)
  - Robots that can learn

How I got here

Many factors, many routes.
But one of them is: School “robotics club”.

How I got here

Many factors, many routes.
But one of them is: School “robotics club”.

How I got here

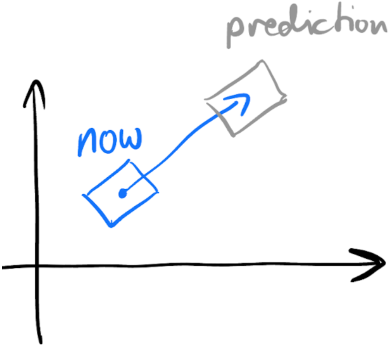
Many factors, many routes.
But one of them is: School "robotics club".

- Robotic hoover $\implies$ navigation

How I got here

Many factors, many routes.
But one of them is: School “robotics club”.

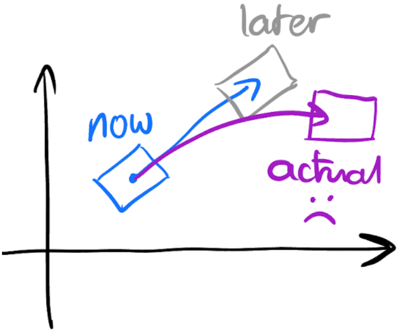
- Robotic hoover $\Longrightarrow$ navigation
- Dead-reckoning



How I got here

Many factors, many routes.
But one of them is: School "robotics club".

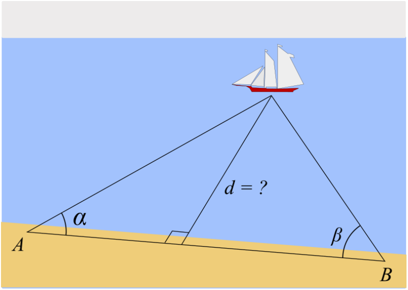
- Robotic hoover $\implies$ navigation
- Dead-reckoning



How I got here

Many factors, many routes.
But one of them is: School “robotics club”.

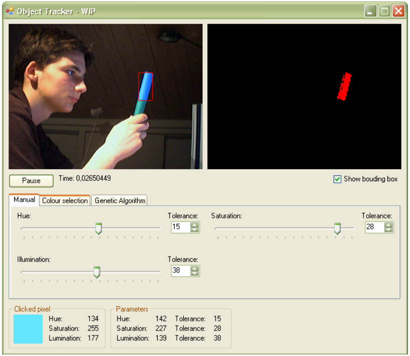
- Robotic hoover $\Longrightarrow$ navigation
- Dead-reckoning
- Triangulation



How I got here

Many factors, many routes.
But one of them is: School "robotics club".

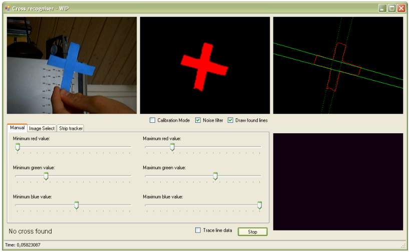
- Robotic hoover $\implies$ navigation
- Dead-reckoning
- Triangulation
- Hough transform



How I got here

Many factors, many routes.
But one of them is: School “robotics club”.

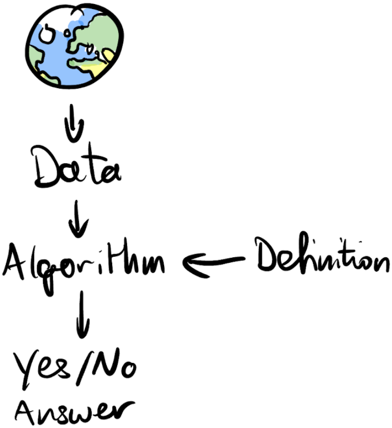
- Robotic hoover $\Longrightarrow$ navigation
- Dead-reckoning
- Triangulation
- Hough transform



How I got here

Many factors, many routes.
But one of them is: School "robotics club".

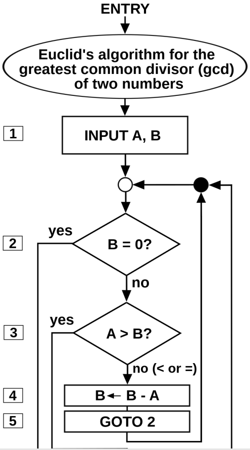
- Robotic hoover $\Longrightarrow$ navigation
- Dead-reckoning
- Triangulation
- Hough transform
- Algorithmic approach



How I got here

Many factors, many routes.
But one of them is: School "robotics club".

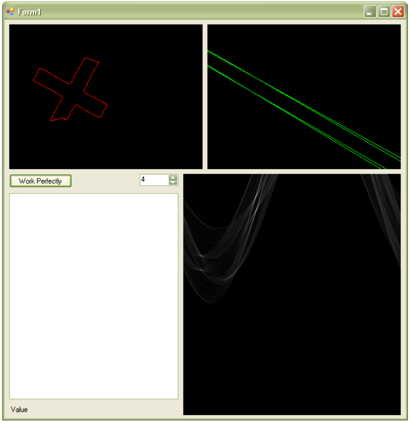
- Robotic hoover $\implies$ navigation
- Dead-reckoning
- Triangulation
- Hough transform
- Algorithmic approach



How I got here

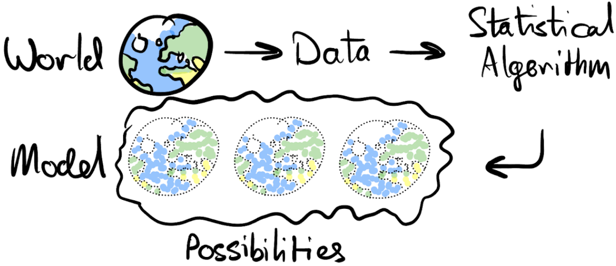
Many factors, many routes.
But one of them is: School "robotics club".

- Robotic hoover $\implies$ navigation
- Dead-reckoning
- Triangulation
- Hough transform
- Algorithmic approach
- Didn't always work
    - If lines arranged 90 degrees from each other...
    - What is a line?
    - What is a cross?
    - Linear reasoning is too rigid for perception.



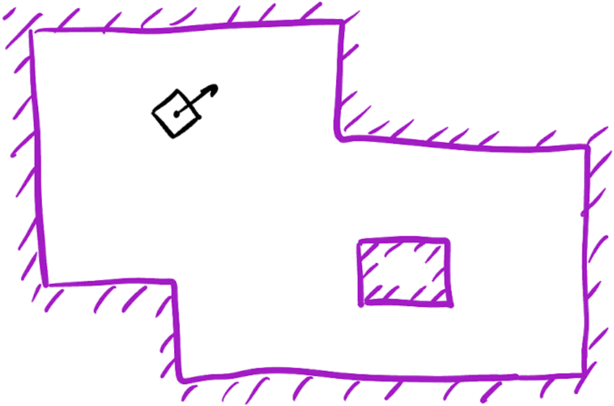
University

- At university, I found out I was taking the wrong approach!
- Data doesn't tell you what is going on with complete **certainty**
  - Are points near each other really in a line?
  - If you can't recognise lines reliably, how do you know if lines are in a cross?
- You need a theory of **evidence accumulation** $\implies$ statistics
  - What are all possible possibilities?
  - Which possibilities are consistent with the data?
  - Keeping track of possibilities allows you to be much more reliable



Particle filter: Accumulating Evidence Example I

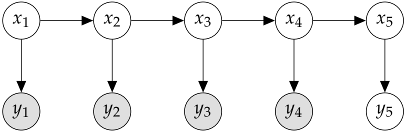
- Back to robot navigation
- Dead reckoning, but with errors (probability distribution!)
- Also has a sensor for where the wall is



Particle filter: Bayesian network

So how did we solve this problem?

- Specify the structure of the problem. Can represent as a **network**:

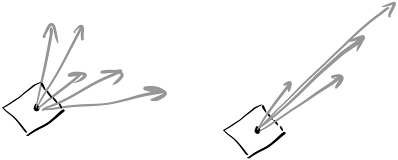


- We understood the probabilities distributions of what could happen: $p(x_t|x_{t-1})$, $p(y_t|x_t)$.
- We could figure out our belief over what was going on.
- Over time, as we got data, our belief became accurate.

But what if we don't know how the world works?

Machine Learning

- It could be that we know a rough structure of the world (the network)
- but not exactly probability of all the things that can happen
- Example: Unknown dead-reckoning accuracy

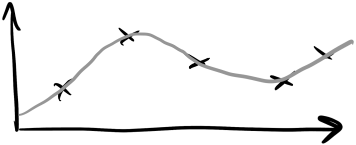


- $\implies$ Use data to find out

Machine Learning

Machine Learning is Curve fitting

We want to learn some input-output relationship:



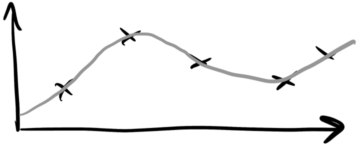
- We specify how to compute the output of such a "function"

$$f_{\mathbf{w},\boldsymbol{\theta}}(x) = \phi_1(x)w_i + \phi_2(x)w_2 + \dots \tag{1}$$

- By adjusting w_i, we can change the shape of the relationship.
- Can adjust the relationship to match data that we get

Machine Learning is Curve fitting

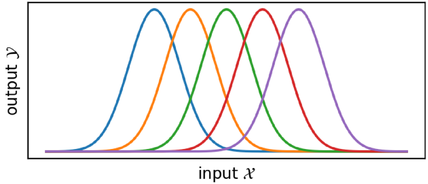
We want to learn some input-output relationship:

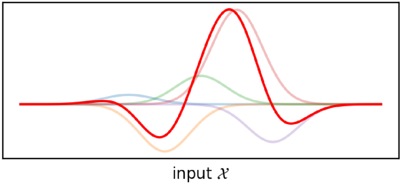


- We specify how to compute the output of such a "function"

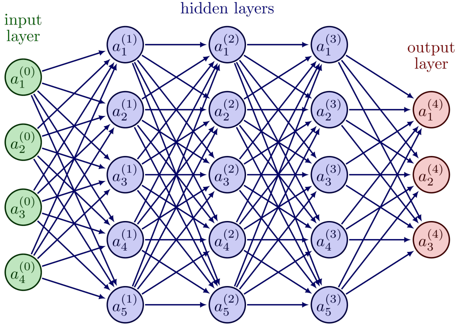
$$f_{\mathbf{w},\boldsymbol{\theta}}(x) = \phi_1(x)w_i + \phi_2(x)w_2 + \dots \qquad (1)$$

- By adjusting w_i, we can change the shape of the relationship.
- Can adjust the relationship to match data that we get





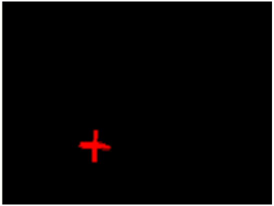
Neural Network fit curves



- A neural network is also just a function, with adjustable weights
- But the function is split up into different smaller parts
- Parts are connected in a **network**
- The network structure must fit the structure of the world (somehow)

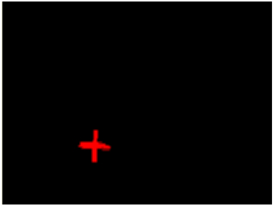
Neural Networks: Recognising Crosses

- Collect lots of examples of images (inputs)
- Label presence of cross (outputs)
- Adjust weights to fit data.



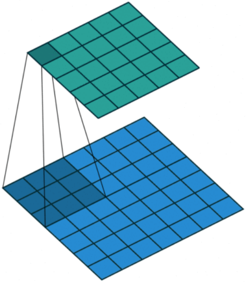
Neural Networks: Recognising Crosses

- Collect lots of examples of images (inputs)
- Label presence of cross (outputs)
- Adjust weights to fit data.
- But what network structure should we use?



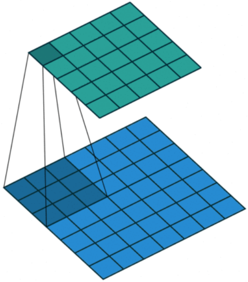
Neural Networks: Recognising Crosses

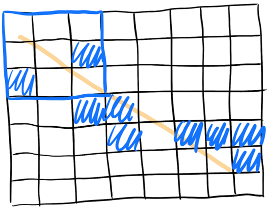
- Collect lots of examples of images (inputs)
- Label presence of cross (outputs)
- Adjust weights to fit data.
- But what network structure should we use?
- People have come up with **convolutions**



Neural Networks: Recognising Crosses

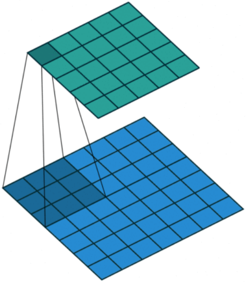
- Collect lots of examples of images (inputs)
- Label presence of cross (outputs)
- Adjust weights to fit data.
- But what network structure should we use?
- People have come up with **convolutions**
- Based on a graph of the problem structure

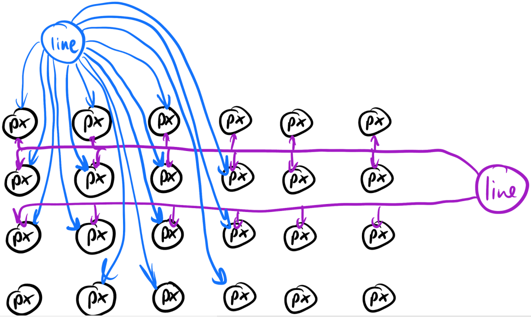




Neural Networks: Recognising Crosses

- Collect lots of examples of images (inputs)
- Label presence of cross (outputs)
- Adjust weights to fit data.
- But what network structure should we use?
- People have come up with **convolutions**
- Based on a graph of the problem structure



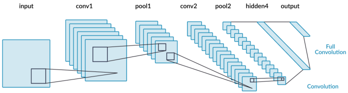


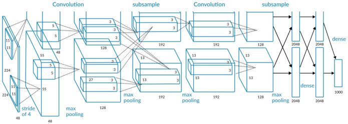
Neural Networks: One Big Question

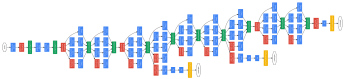
- Every time a NN is used, a **network** needs to be designed.

Neural Networks: One Big Question

- Every time a NN is used, a **network** needs to be designed.
- Architectures are getting more complex. Very laborious!

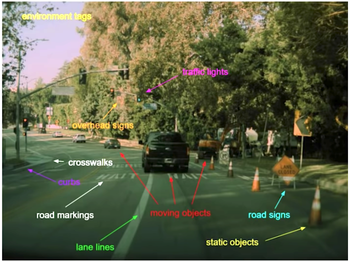






Neural Networks: One Big Question

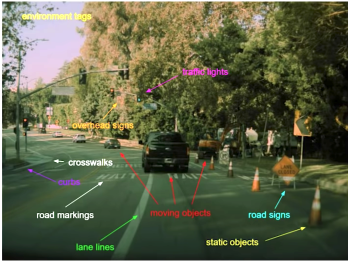
- Every time a NN is used, a **network** needs to be designed.
- Architectures are getting more complex. Very laborious!
- We are not learning the most important things: Concepts!



Neural Networks: One Big Question

- Every time a NN is used, a **network** needs to be designed.
- Architectures are getting more complex. Very laborious!
- We are not learning the most important things: Concepts!

Main tool is **trial-and-error**
("cross-validation")

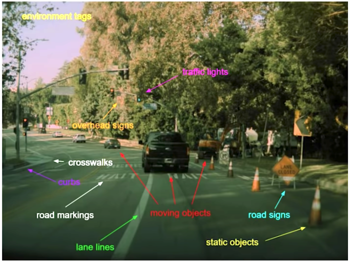


Neural Networks: One Big Question

- Every time a NN is used, a **network** needs to be designed.
- Architectures are getting more complex. Very laborious!
- We are not learning the most important things: Concepts!

Main tool is **trial-and-error**
("cross-validation")

Goal: Learn network connectivity automatically!



Where I am now

- I got my tenure!
- Academia talk for "won't get fired if I continue what I'm doing"
- "Vast contract"
- Best thing about this:
  Working with a team of brilliant PhD students, on these very problems.


Anish Dhir


Artem Artemev


Jose Pablo Folch


Ruby Sedgwick


Seth Nabarro


Tycho van der Ouderaa

Where from here?

Broadly speaking I think that biological neurons can do things that computer neurons cannot yet:

- Learn connectivity structure of the network.
- Learn while only communicating with nearby neurons.

Measuring uncertainty
can help solve both these problems

Marginal likelihood computation in action

Marginal likelihood in action

Marginal likelihood in action

Conclusion

- We need uncertainty to make sense about the world
- Given a network structure, we can figure out what happened (deduction)
- The big goal is to find the network structure (induction)
- There's so much more interesting things I could talk about

but for now...

Thanks for listening.