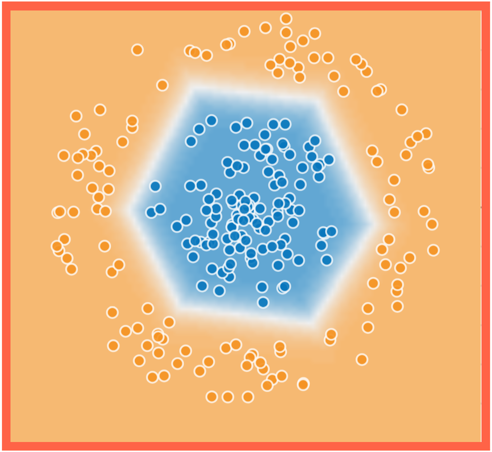
What is...machine learning in mathematics - part 6?

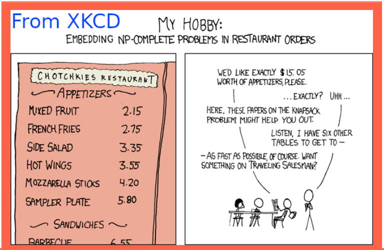
Or: Approximating hard problems

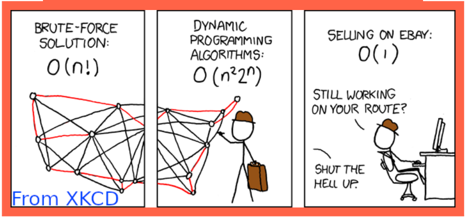
Neural network (NN) = approximation



- Above A NN finds a polygon approximation of a circle
- Essentially a NN is a way to approximate a potentially complicated function
- A NN approximates by combining simple functions into a complex model

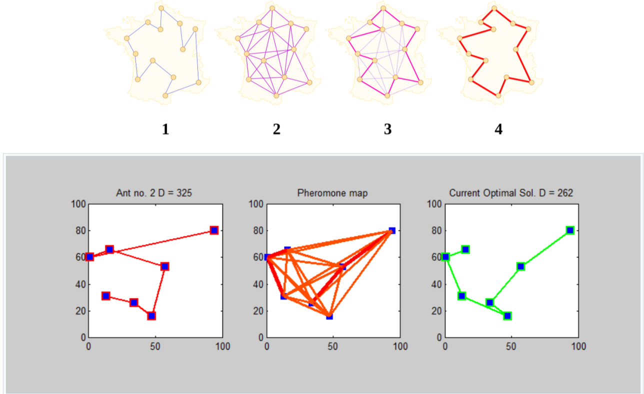
Hard problems





- For this video define hard to be ‘Any algorithm for computation is nasty’
- Example NP-complete problems ‘$=$’ not solvable in polynomial runtime
- Hard problems often require heuristic or approximate solutions

Traveling salesperson problem (TSP)



Ant colony optimization algorithm for a TSP with 7 cities: Red and thick lines in the pheromone map indicate presence of more pheromone

- **TSP** Find the shortest route visiting each city exactly once and returning home
- **This is hard** Fastest known algorithm needs $\approx 2^{\#cities}$ operations
- **Idea** Use NN to approximate a near-optimal solution efficiently

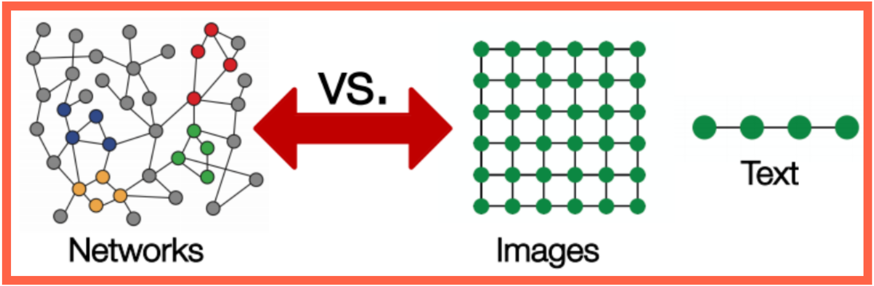
Enter, the theorem

A 2018 GNN (https://arxiv.org/abs/1809.02721) can predict the TSP tour with

high probability and low derivation

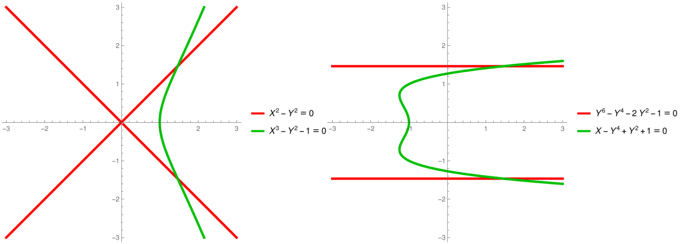
Roughly, 80% probability on a 2% derivation interval

- Very impressive This is close to results heuristics for TSP obtained by 'traditional methods', e.g. multi-fragment algorithms
- Graph NN (GNN) is a generalization of a convolutional NN (CNN)



Promising!

The same intersection set in two different ways



Question. How can we algebraically see that the intersections match?

$$(X^2 - Y^2, X^3 - Y^2 - 1) \stackrel{?}{=} (Y^6 - Y^4 - 2Y^2 - 1, X - Y^4 + Y^2 + 1)$$

- There are actually many other examples along the same lines
- Explicitly Computation of Gröbner bases is EXPSPACE-hard (doubly exponential in the worst case)
- Great A NN can do this efficiently, too!

Thank you for your attention!

I hope that was of some help.