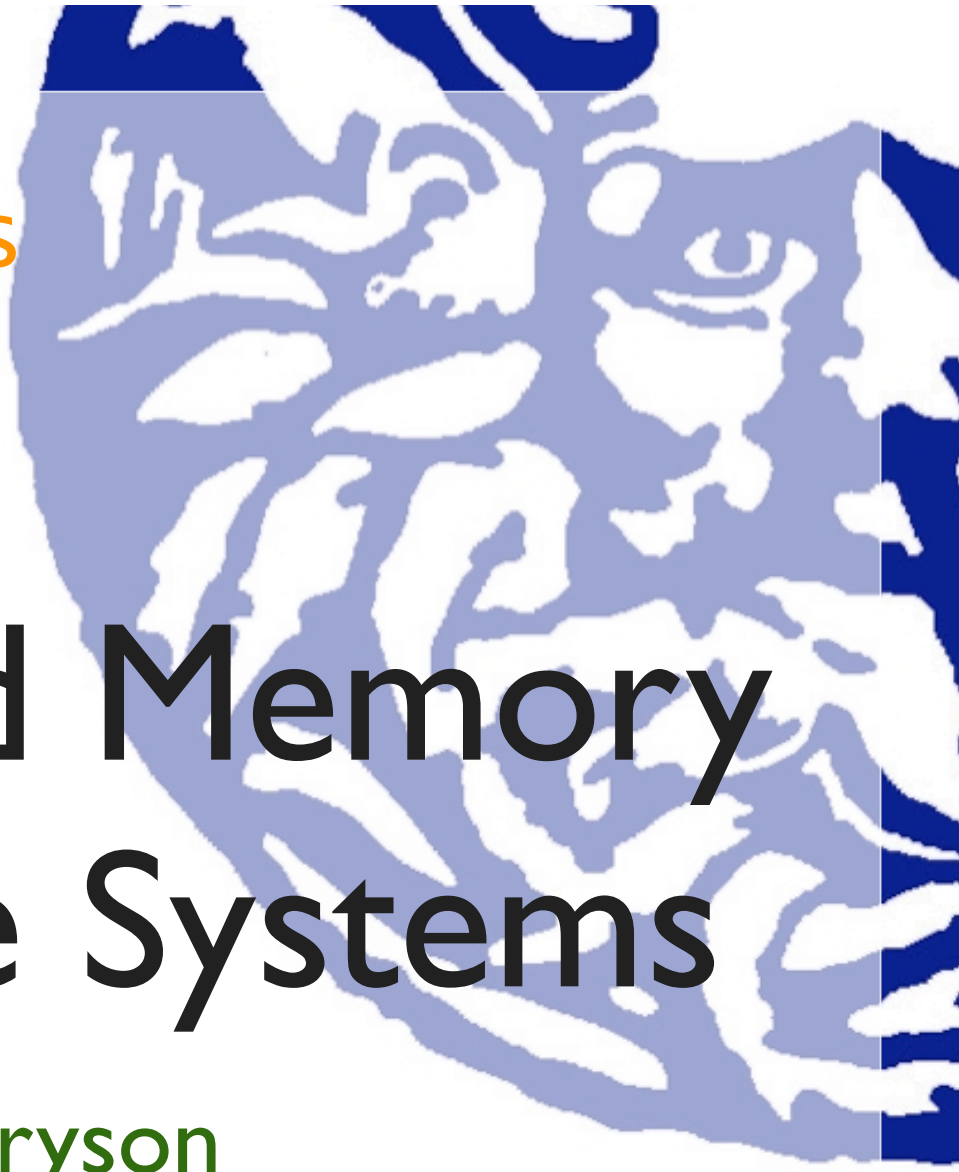
Intelligent Control and Cognitive Systems

brings you...



Learning and Memory in Cognitive Systems

Joanna J. Bryson

University of Bath, United Kingdom

Sensing vs Perception

- First week: Sensing – what information comes in.
- This week: Perception – what you think is going on.
  - Perception includes expectations.
  - Necessary for disambiguating noisy and impoverished sensory information.

Bayes' Theorem

"expectations"

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

$$p(X) = \sum_{Y} p(X|Y)p(Y)$$

posterior ∝ likelihood × prior

Given you've seen X, you can figure out if Y is likely true based on what you already know about the probability of experiencing: X independently, Y independently and X when you see Y.

note to JB: copy X,Y to board, useful later

One Application...

- Y – potential action
- X – sensing
- priors = memory
- priors + sense = perception

Expectations

- For all cognitive systems, some priors are hard-coded: body shape, sensing array, even neural connectivity.
  - Derived from the experience of evolution or from a designer.
- Other expectations are derived from an individual's own experience – learning.

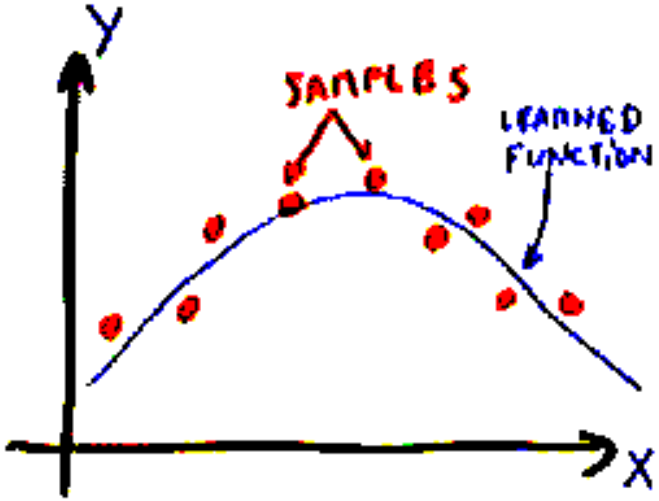
Learning

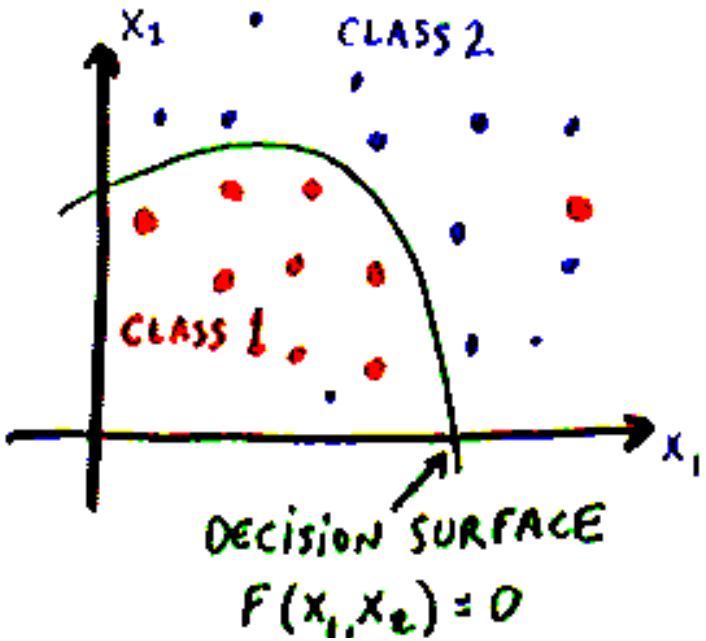
- Learning requires:
  - A representation.
  - A means of acting on current evidence.
  - A means of incorporating feedback concerning the outcome of the guess.
- AI learning calls incorporating feedback "error correction".

Learning Outcomes

- Learning is not just memorisation!
- Objective is to do the right thing at the right time (to be intelligent.)
- Doing the right thing often requires predicting likely possible sensory conditions so you can disambiguate situations that would otherwise be perceptually aliased.
- Prediction is done by generalising previous experience.

Two Kinds of Supervised Learning





What we'll use as an example today.

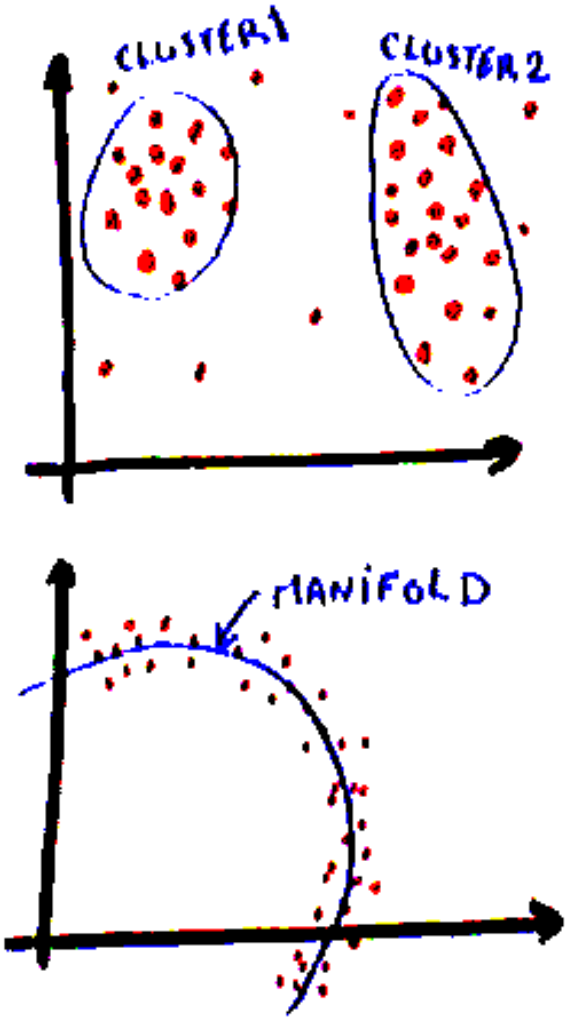
- Regression: also known as "curve fitting" or "function approximation". Learn a continuous input-output mapping from a limited number of examples (possibly noisy).
- Classification: outputs are discrete variables (category labels). Learn a decision boundary that separates one class from the other. Generally, a "confidence" is also desired (how sure are we that the input belongs to the chosen category).

Includes kernel methods (not covered here.)

Unsupervised Learning

c.f. Lecture 5 "what the brain seems to be doing"

Unsupervised learning comes down to this: if the input looks like the training samples, output a small number, if it doesn't, output a large number.



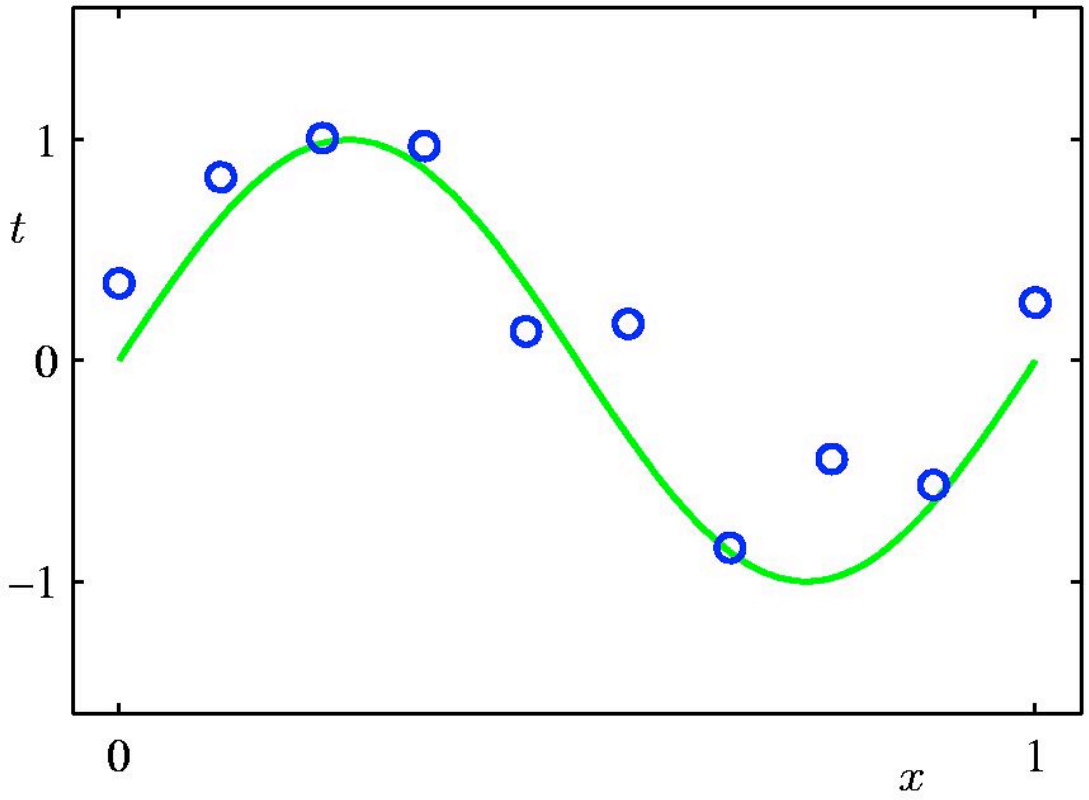
- This is a horrendously ill-posed problem in high dimension. To do it right, we must guess/discover the hidden structure of the inputs. Methods differ by their assumptions about the nature of the data.
- A Special Case: Density Estimation. Find a function f such $f(X)$ approximates the probability density of X, $p(X)$, as well as possible.
- Clustering: discover "clumps" of points
- Embedding: discover low-dimensional manifold or surface near which the data lives.
- Compression/Quantization: discover a function that for each input computes a compact "code" from which the input can be reconstructed.

Unsupervised?

- Learning requires:
  - A representation.
  - A means of acting on current evidence.
  - A means of incorporating feedback concerning the outcome of the guess.
- The distinction between "supervised" and "unsupervised" learning is fairly arbitrary. There's always some feedback mechanism.
  - Information comes from somewhere: either the representation, the searched domain, or the error signal. cf. No Free Lunch (Wolpert)

Polynomial Curve Fitting

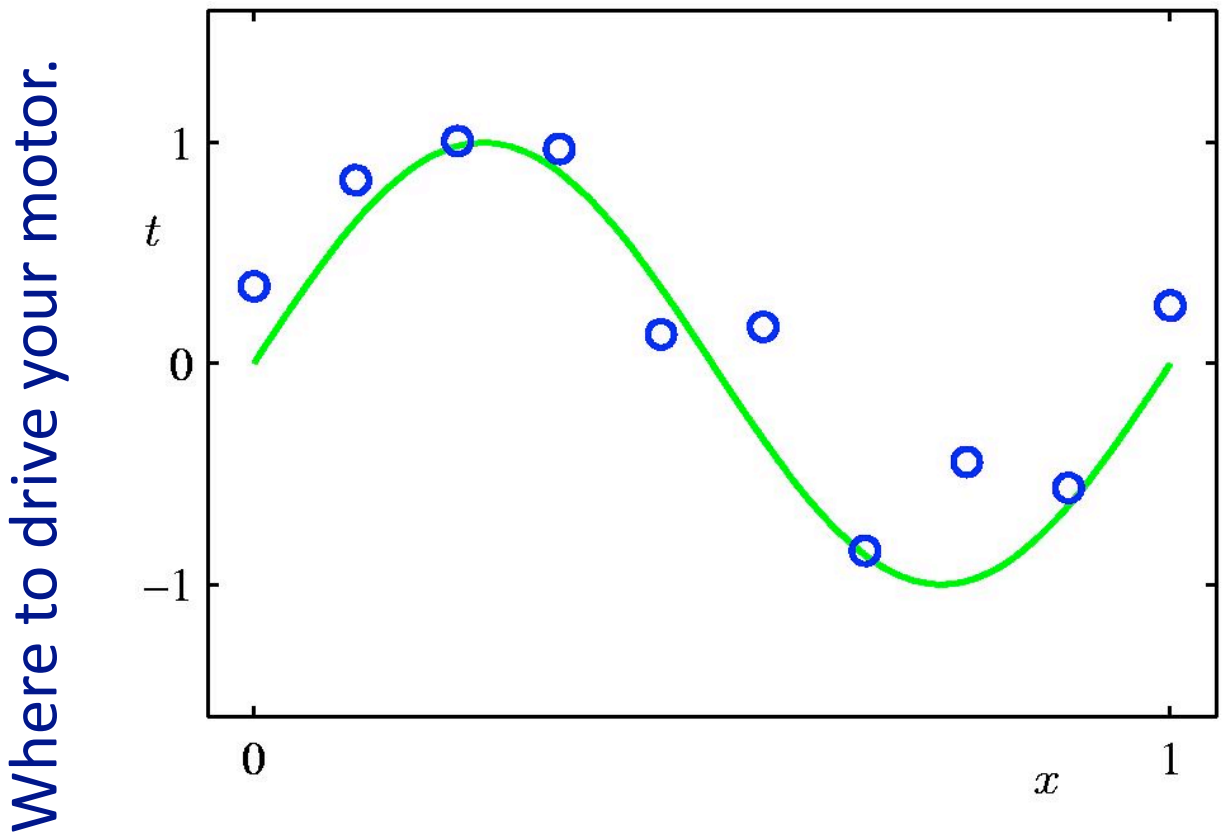
"Regression"
via Chris Bishop



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Representation: Just a polynomial equation.

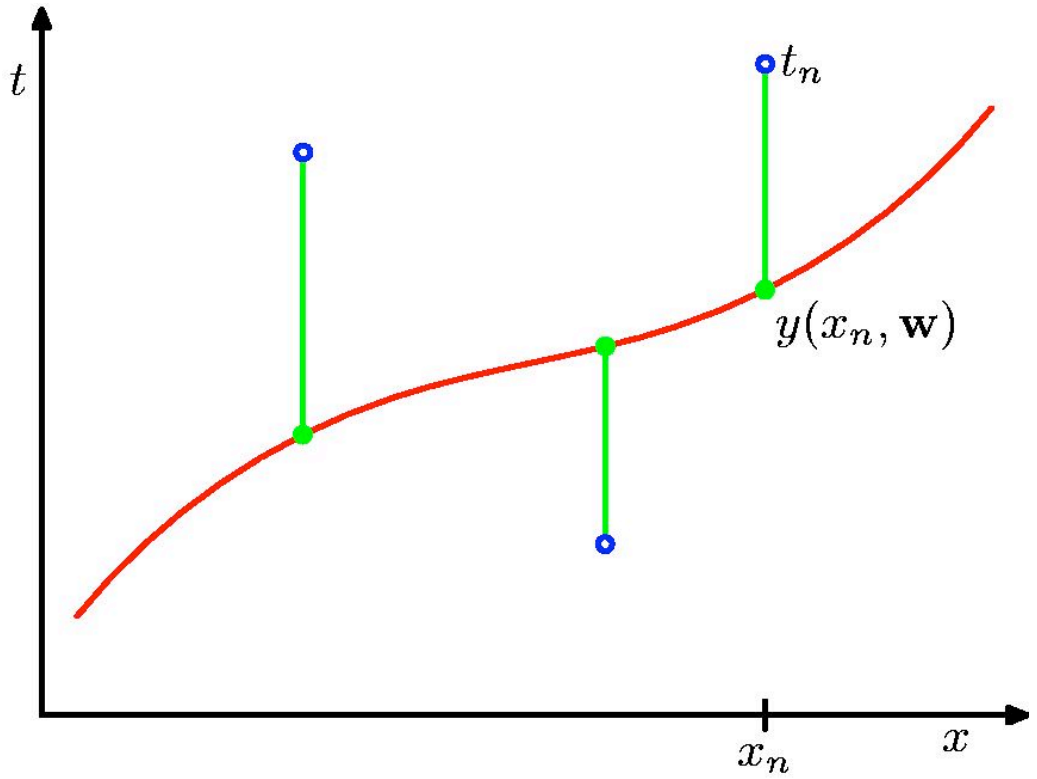
Example Application to Action Selection



What you sensed

Sum-of-Squares Error Function

Use data to fix the world model currently held in the representation.



$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

Error functions

- Based on some parameter **w** (for *weight* – more on why it's called that later).
- Objective is to minimise error function.
  - Take its derivative with respect to w.
  - Go down (take second deriv. if nec.)
- Linear functions gives a nice U function ∴ you can tell when your done, derivative = 0.

Theory vs Practice

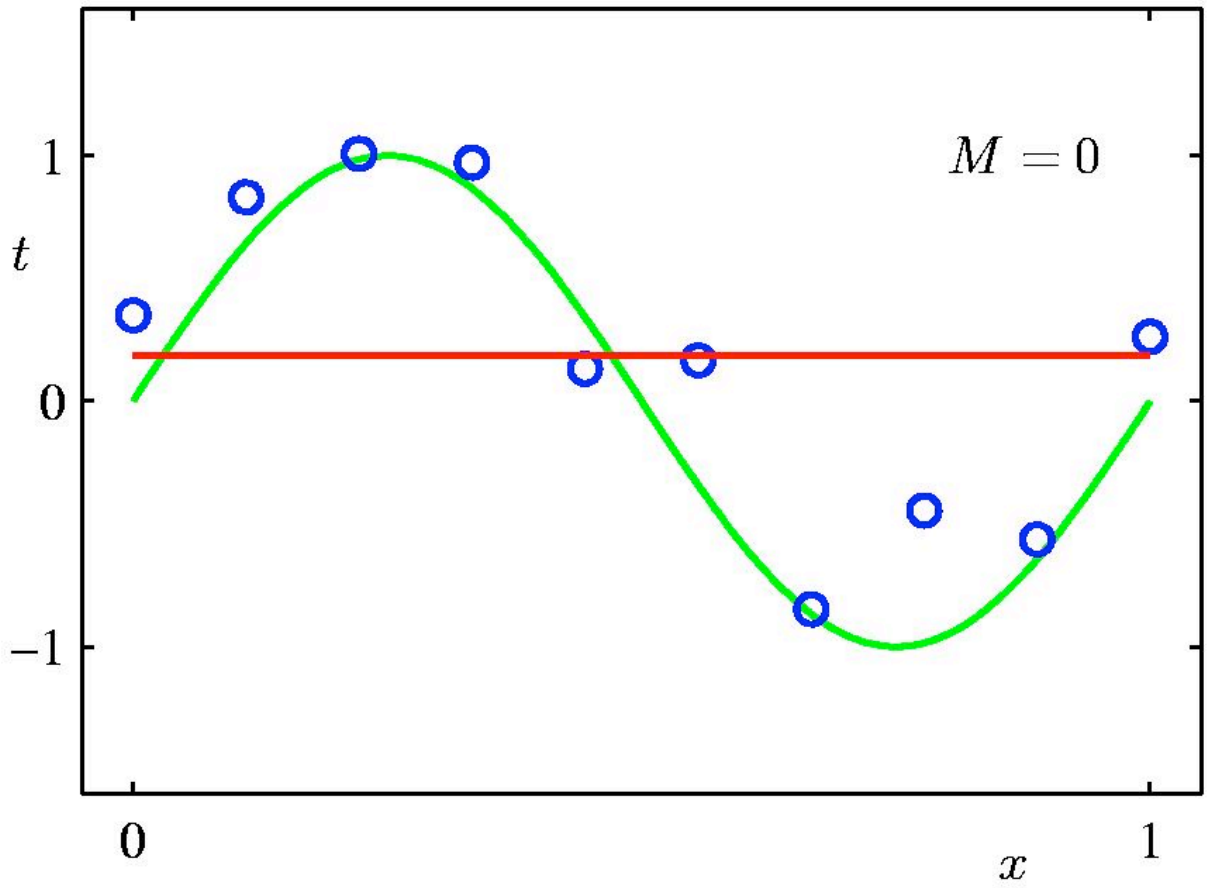
- If we assume that noise in signal is Normally distributed (with fixed variance), then least squares is equivalent to probabilistic methods (Per CM20220).
- Least squares is a lot easier to implement & lighter-weight to run.
- To the extent the assumption doesn't hold, quality of results degrades – may be OK.

Why Representations Matter

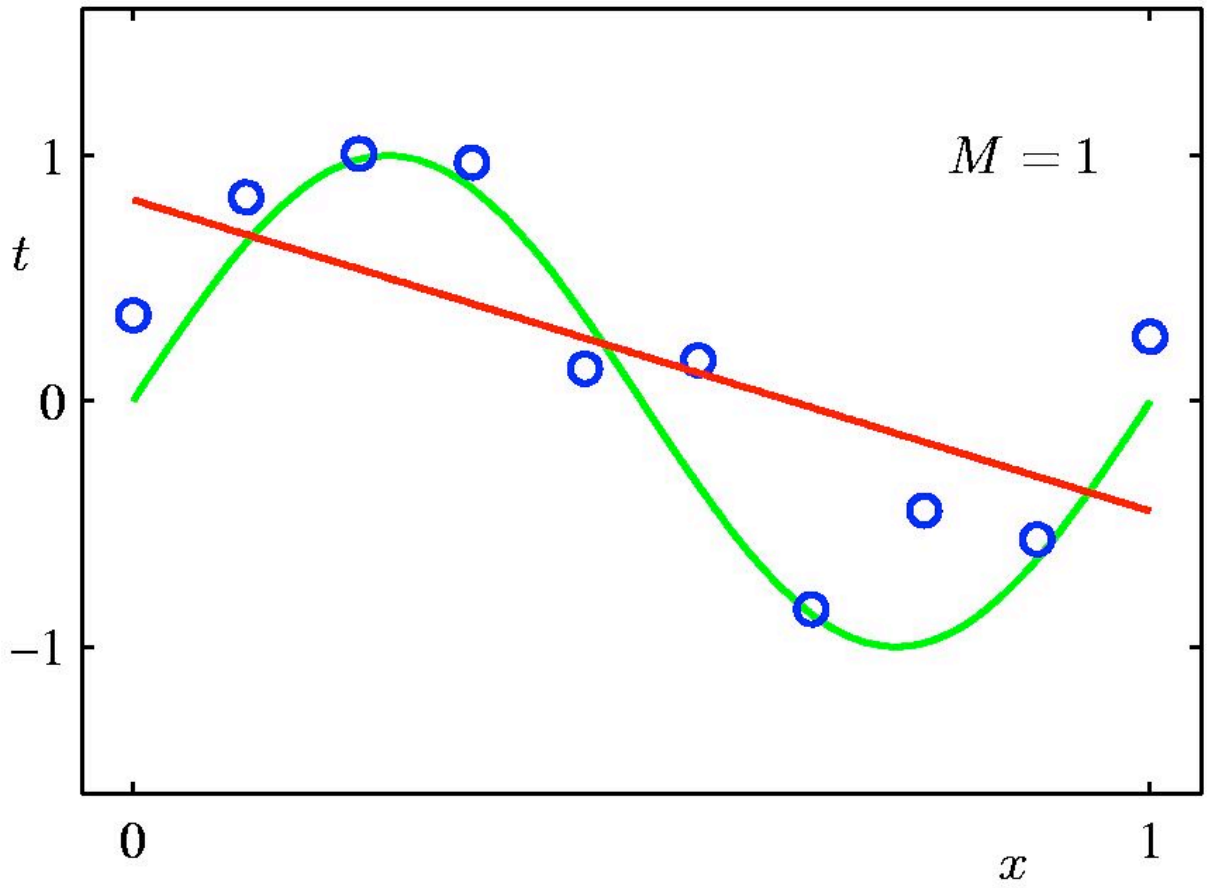
Green line is model used to generate data (in combination with noise).

Red line is the model learned from observing that data.

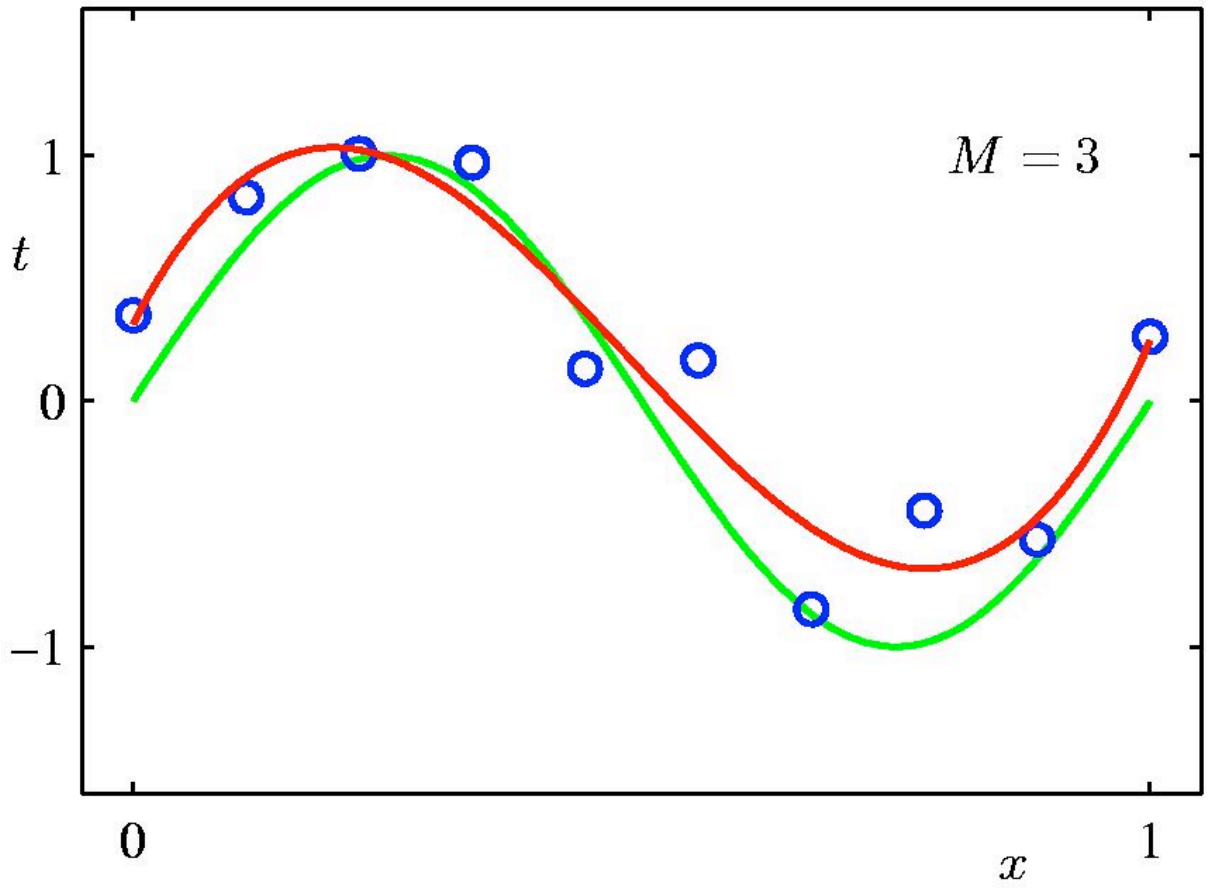
0th Order Polynomial



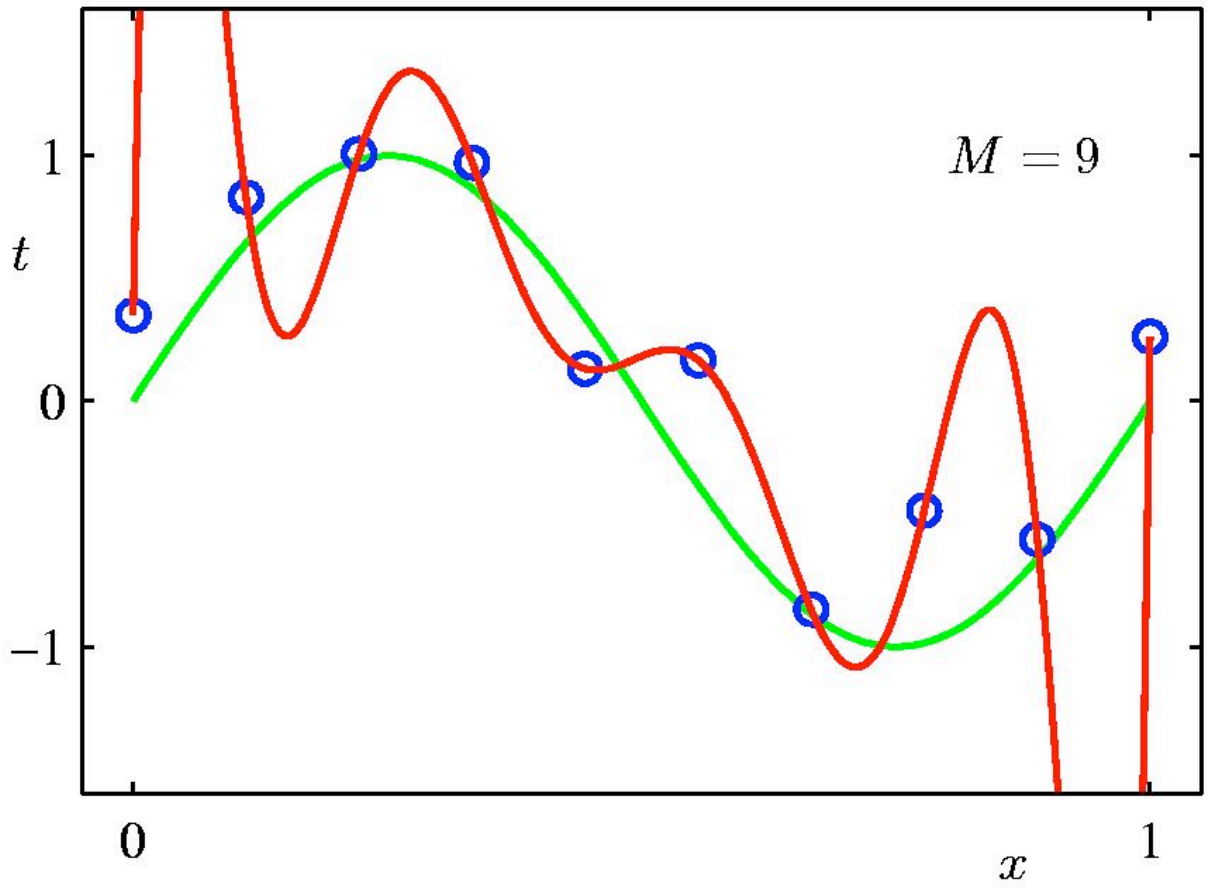
1st Order Polynomial



3rd Order Polynomial

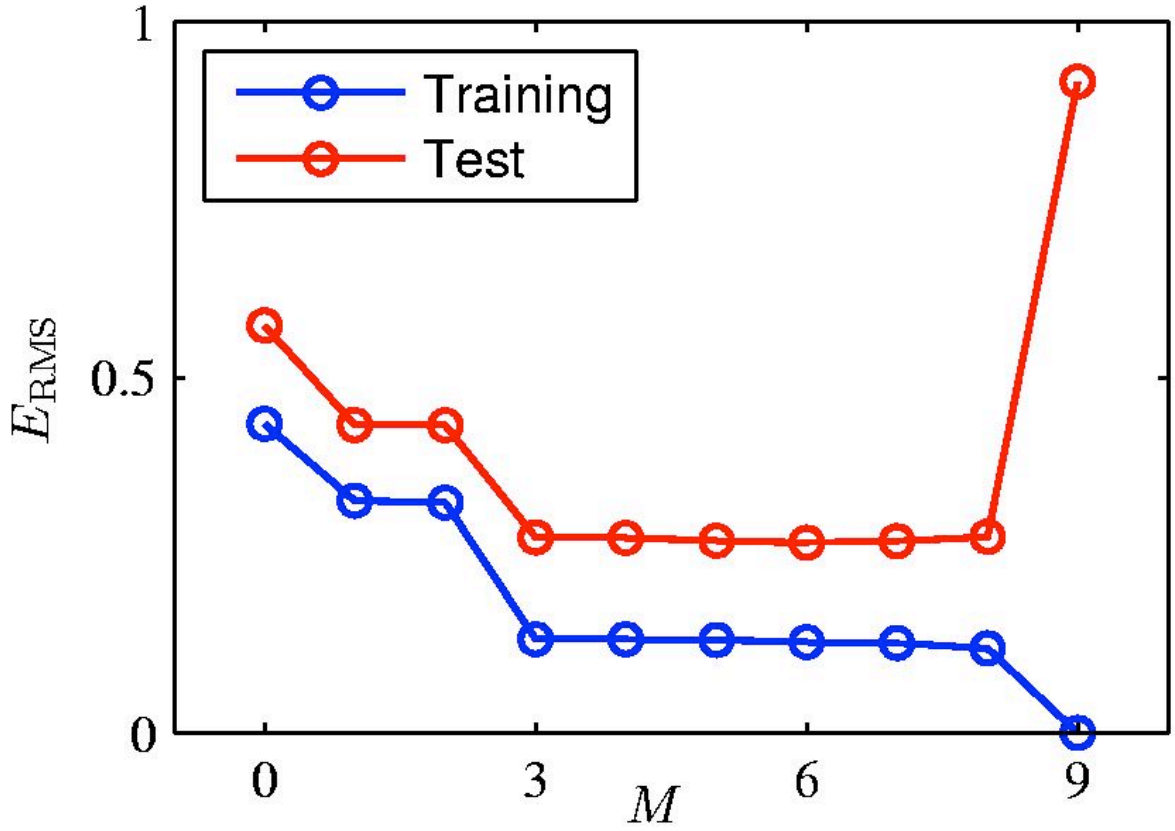


9th Order Polynomial



Over-fitting

When your model is too powerful for the data, it just "rote memorises" without generalising.



That means you get better on training data but worse on data you haven't seen.

Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

Polynomial Coefficients

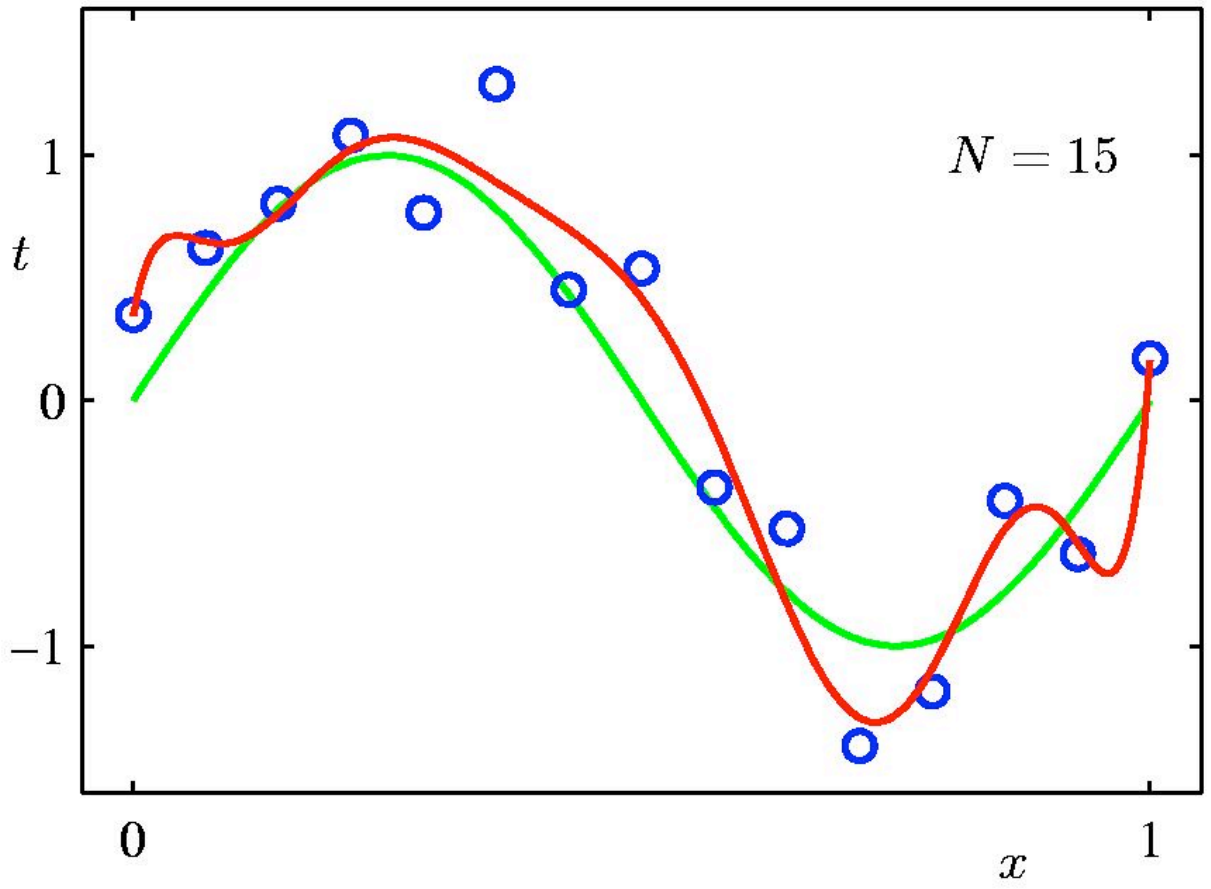
Spot an indication of a problem.

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

More data makes it better...

Data Set Size: $N = 15$

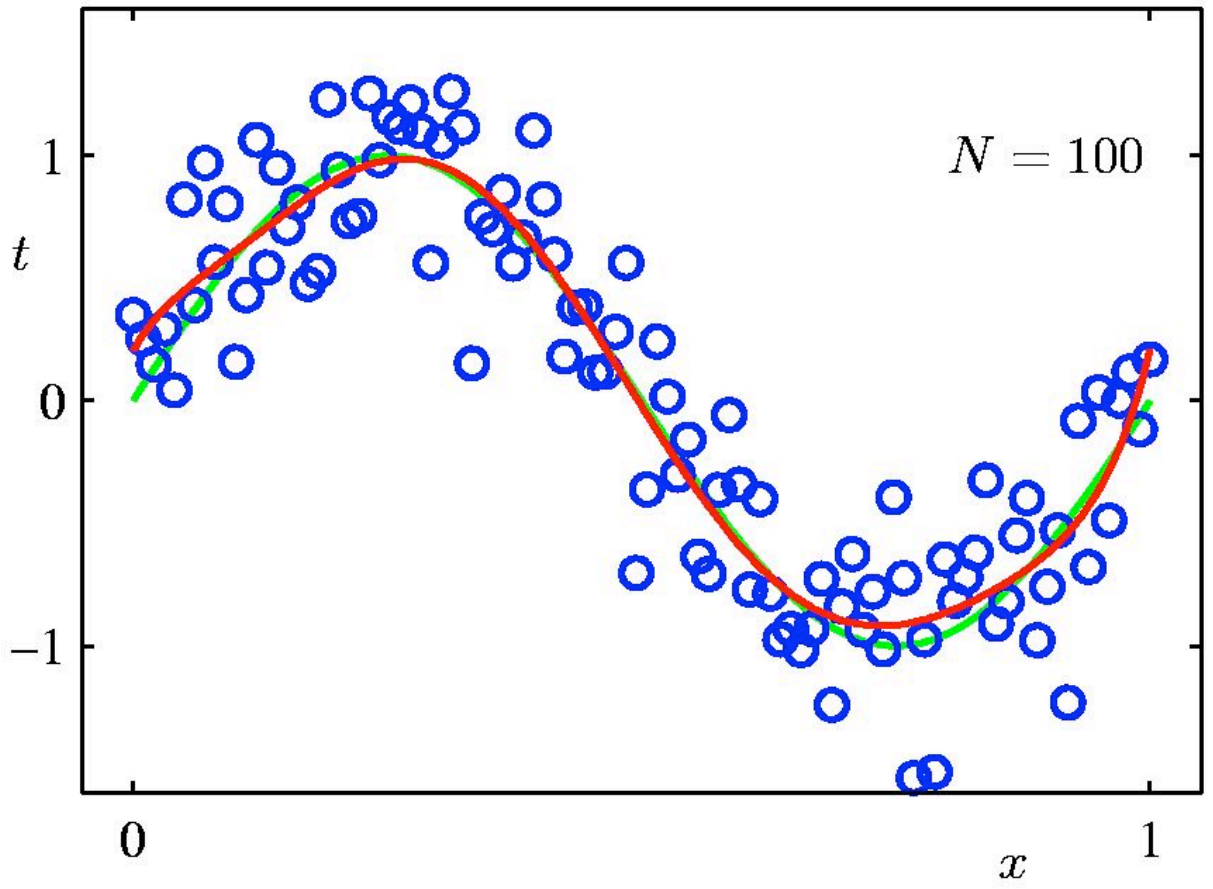
9th Order Polynomial



Data Set Size: $N = 100$

and better... more data is more information on the underlying model!

9th Order Polynomial



Overfitting

- If you can memorise everything then you have no error signal to learn from, so you can't improve your model.
- If you can really memorise everything this doesn't matter. "Generalisation isn't the point of learning. Being right is the point of learning." – Will Lowe
- But mostly, it matters.

Polynomial Coefficients

Spot the indication of a problem.

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

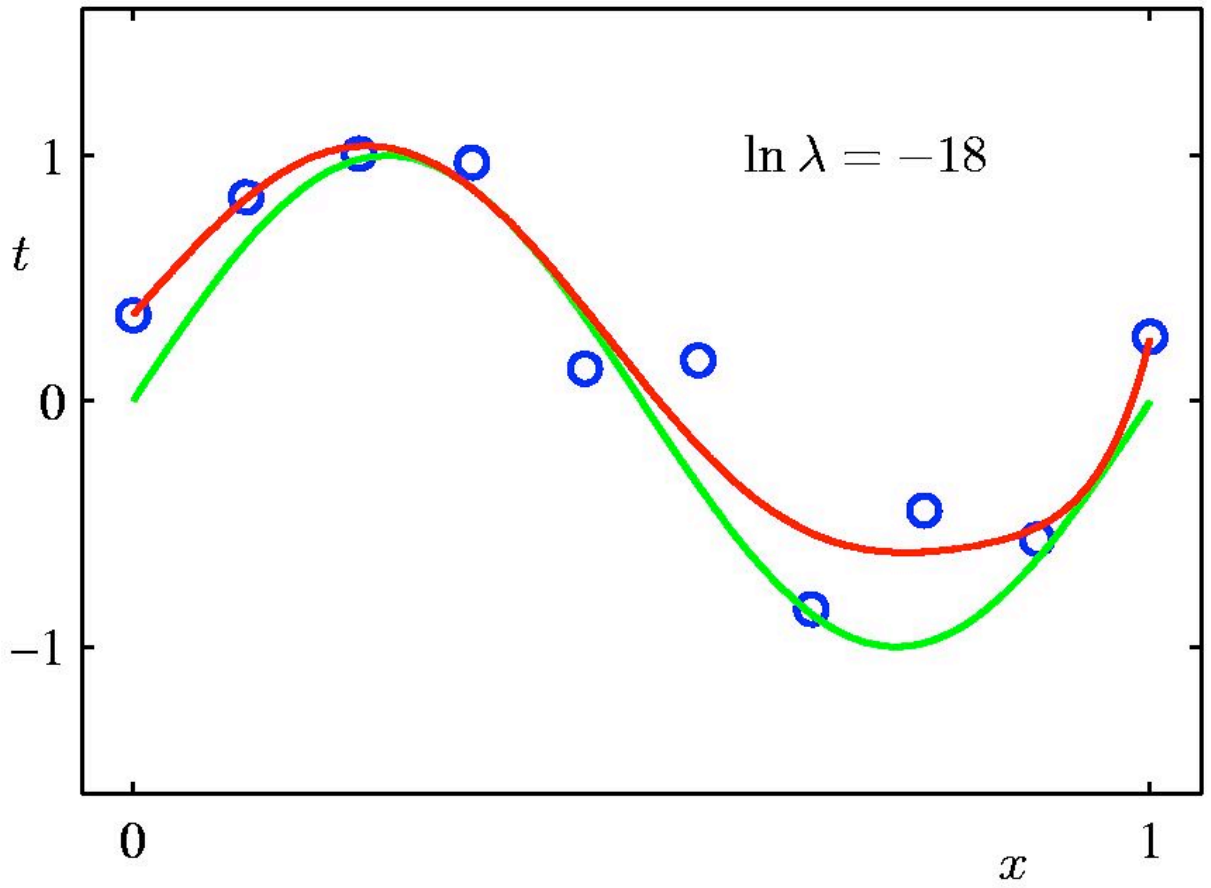
Regularization

Another solution (when you can't get more data)

Penalize large coefficient values

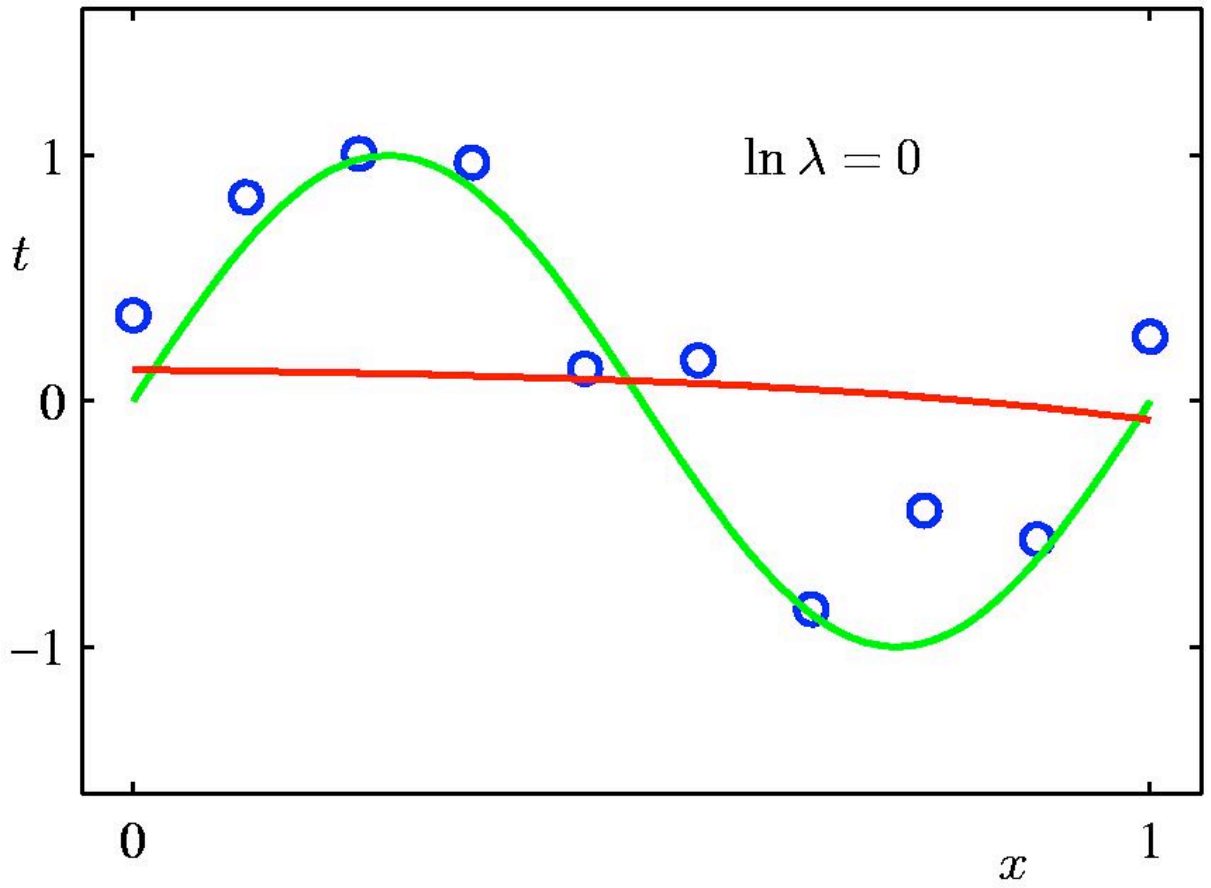
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Regularization: $\ln \lambda = -18$



Regularization: $\ln \lambda = 0$

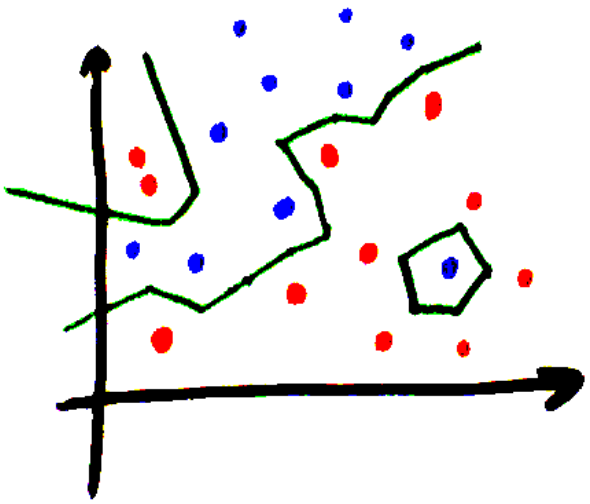
No free lunch! Have to guess the right λ now.



Common Representations

- Production Rules
- Neural Networks
- "Genomes" (for Genetic Algorithms)
- Mixtures of Gaussians
- Vectors (counts) e.g. Word Embeddings
- Many more (many *ad hoc*)

A Simple Trick: Nearest Neighbor Matching

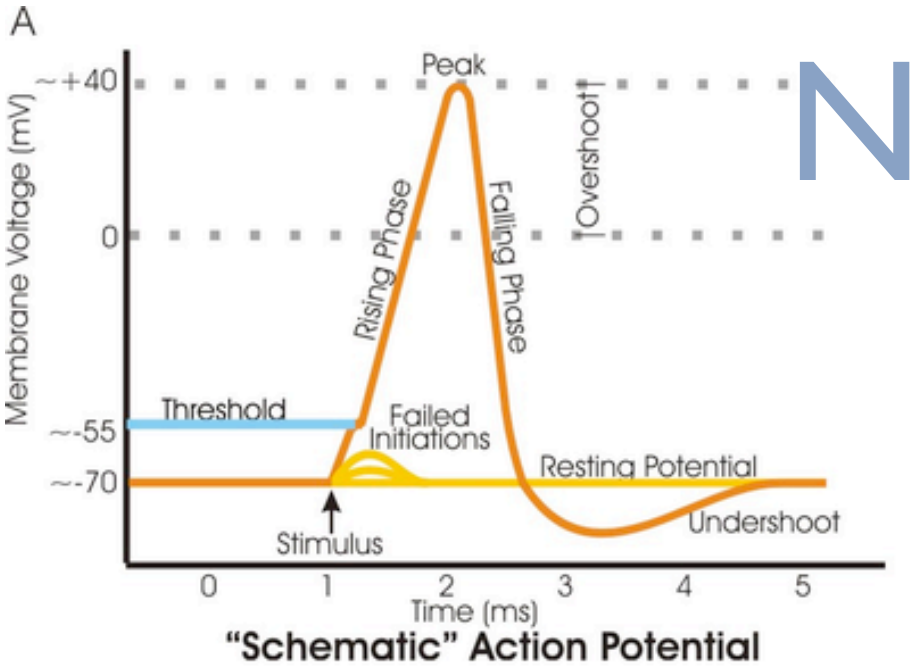


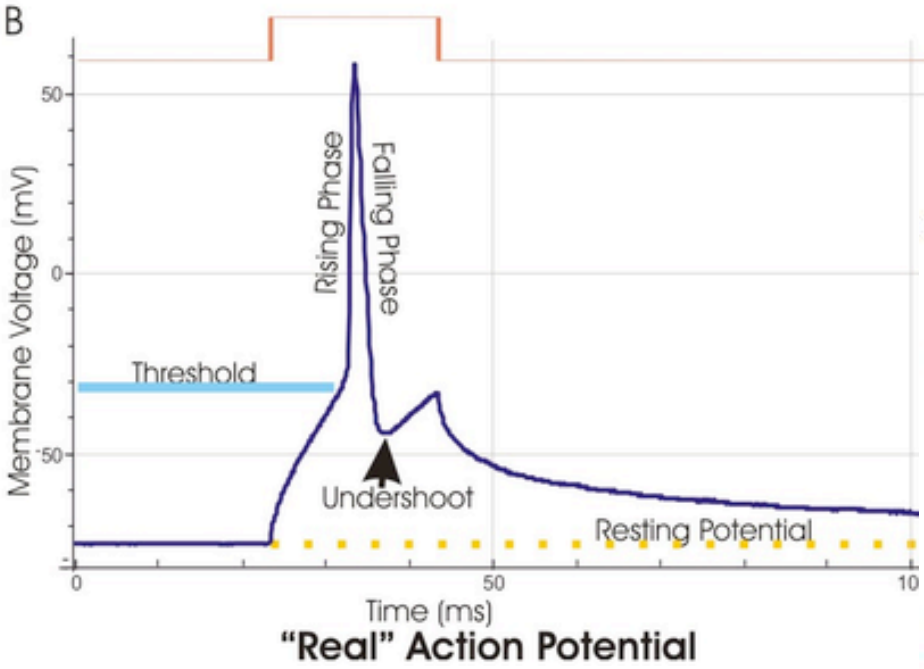
Problem, problem, problem but it works really well.

- Instead of insisting that the input be exactly identical to one of the training samples, let's compute the "distances" between the input and all the memorized samples (aka the prototypes).
- 1-Nearest Neighbor Rule: pick the class of the nearest prototype.
- K-Nearest Neighbor Rule: pick the class that has the majority among the K nearest prototypes.
- PROBLEM: What is the right distance measure?
- PROBLEM: This is horrendously expensive if the number of prototypes is large.
- PROBLEM: do we have any guarantee that we get the best possible performance as the number of training samples increases?

Can often also interpolate between stored solutions (Atkins, Schaal)

Neural Networks

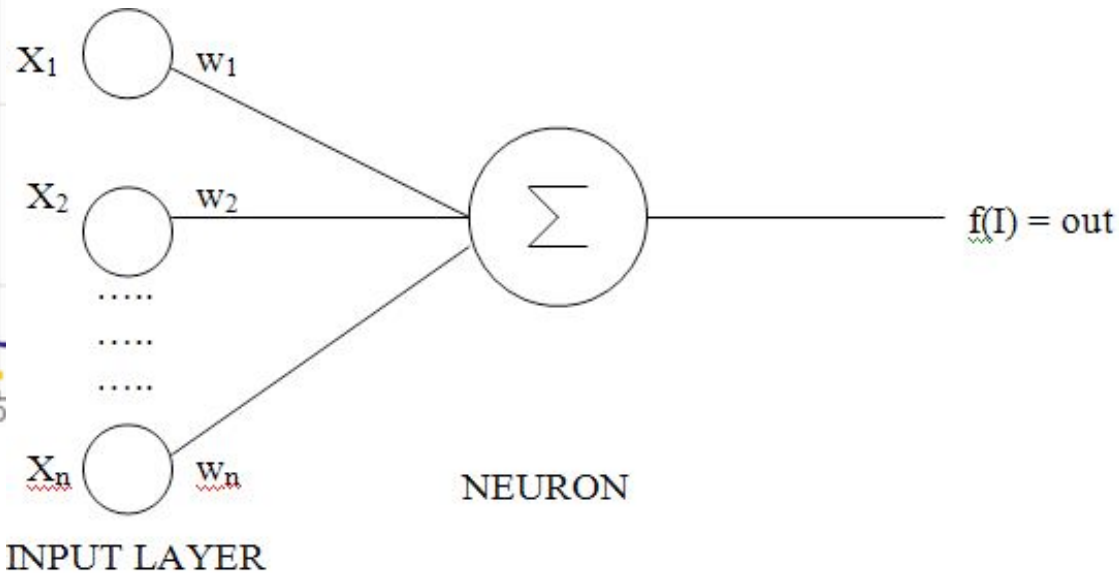




wikipedia

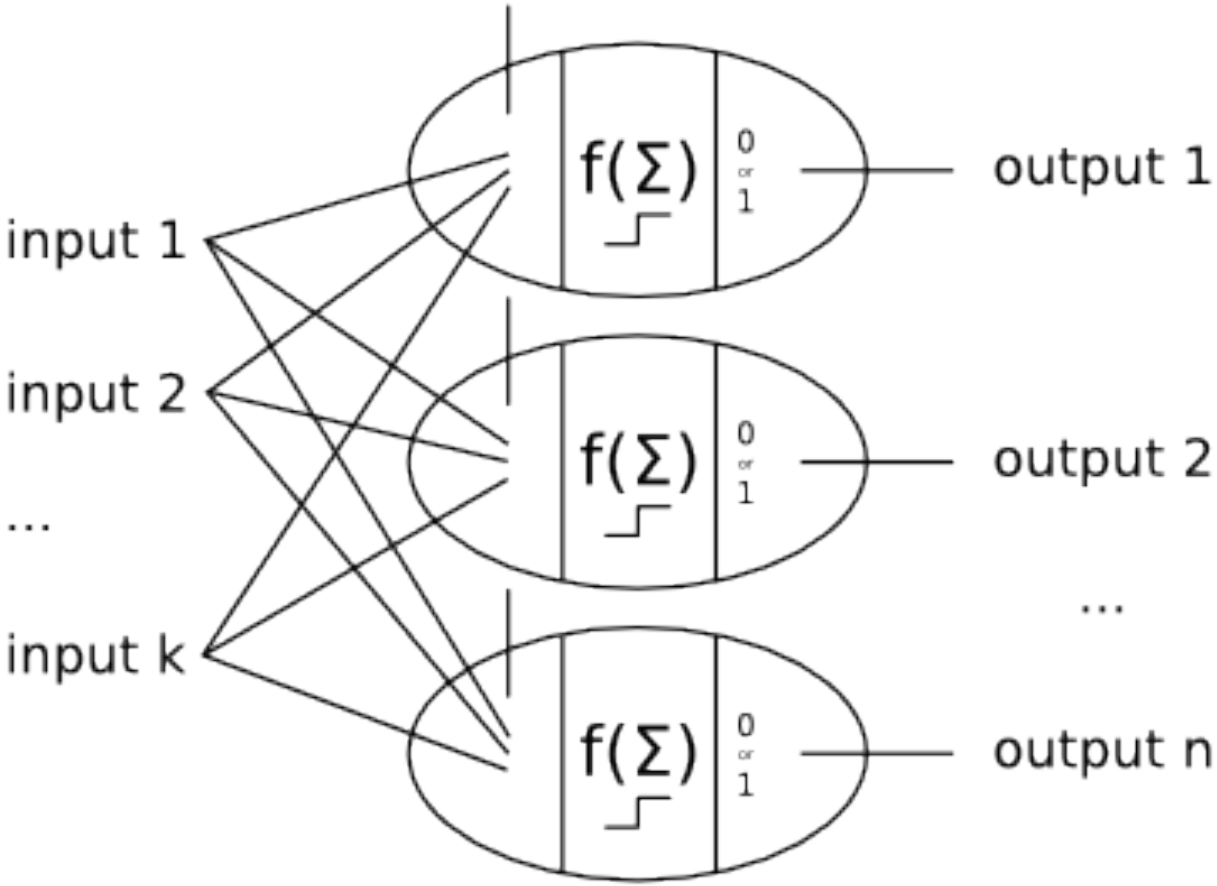
Biomimetic approaches to machine learning date back to the fifties (Rosenblatt 1958).

A perceptron, inspired by neurons.



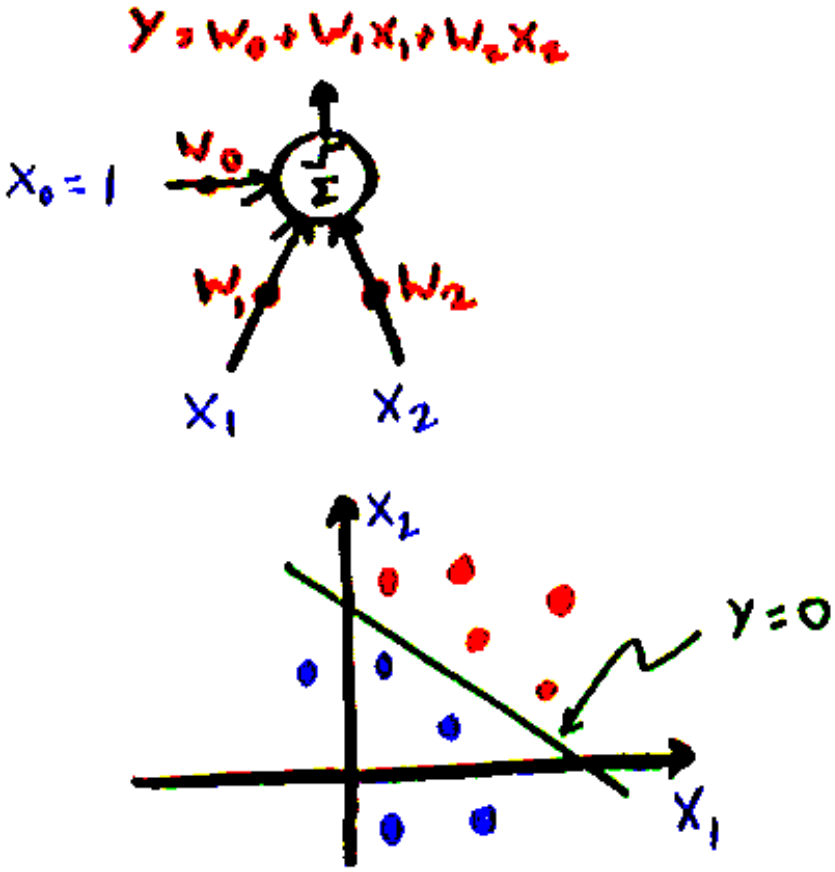
Single Layer Perceptron Network

Note:
mutual inhibition
"winner take all"
WTA



The Linear Classifier (originally: The Perceptron)

Historically, the Linear Classifier was designed as a highly simplified model of the neuron (McCulloch and Pitts 1943, Rosenblatt 1957):



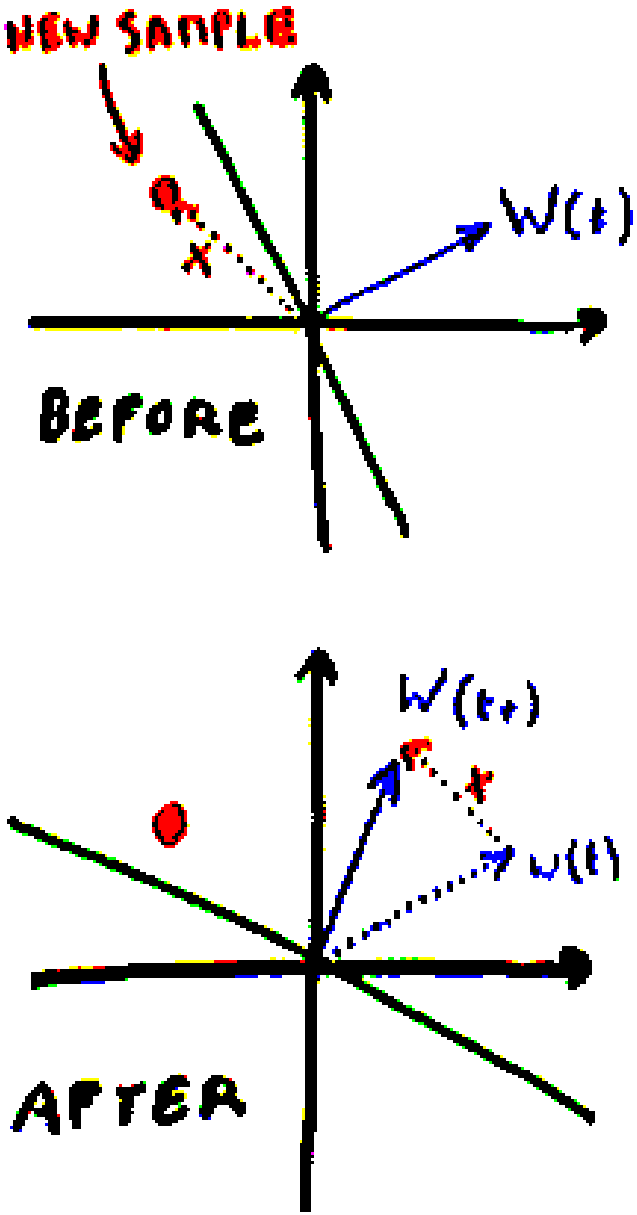
$$y = f(\sum_{i=0}^{i=N} w_i x_i)$$

With f is the threshold function: $f(z) = 1$ iff $z > 0$, $f(z) = -1$ otherwise. x_0 is assumed to be constant equal to 1, and w_0 is interpreted as a bias.
In vector form: $W = (w_0, w_1 .... w_n), X = (1, x_1 ... x_n)$:

$$y = f(W'X)$$

The hyperplane $W'X = 0$ partitions the space in two categories. W is orthogonal to the hyperplane.

A Simple Idea for Learning: Error Correction



Perceptron Learning Algorithm

We have a **training set** $\mathcal{S}$consisting of P input-output pairs: $\mathcal{S} = (X^1, y^1), (X^2, y^2), ...(X^P, y^P)$.
A very simple algorithm:
- show each sample in sequence repetitively
- if the output is correct: do nothing
- if the output is -1 and the desired output +1: increase the weights whose inputs are positive, decrease the weights whose inputs are negative.
- if the output is +1 and the desired output -1: decrease the weights whose inputs are positive, increase the weights whose inputs are negative.

More formally, for sample p:

$$w_i(t+1) = w_i(t) + (y_i^p - f(W'X^p))x_i^p$$

This simple algorithm is called the Perceptron learning procedure (Rosenblatt 1957).

The Perceptron Learning Procedure

Provably works iff linearly separable.

Theorem: If the classes are linearly separable (i.e. separable by a hyperplane), then the Perceptron procedure will converge to a solution in a finite number of steps.
Proof: Let's denote by W^* a normalized vector in the direction of a solution. Suppose all X are within a ball of radius R. Without loss of generality, we replace all X^p whose y^p is -1 by $-X^p$, and set all y^p to 1. Let us now define the margin $M = min_p W^* X^p$. Each time there is an error, $W.W^*$ increases by at least $X.W^* \geq M$. This means $W_{final}.W^* \geq NM$ where N is the total number of weight updates (total number of errors). But, the change in square magnitude of W is bounded by the square magnitude of the current sample X^p, which is itself bounded by R^2. Therefore, $|W_{final}|^2 \leq NR^2$. combining the two inequalities $W_{final}.W^* \geq NM$ and $|W_{final}| \leq \sqrt{N}R$, we have

$$W_{final}.W^*/|W_{final}| \geq \sqrt{(N)}M/R$$

. Since the left hand side is upper bounded by 1, we deduce

$$N \leq R^2/M^2$$

Proof by Minsky (long story, maybe worth reading on wikipedia)

Learning Algorithm Terms & Tricks

- How much you add or subtract from the weight determines how fast you learn: learning rate.
- If you learn too fast you can overshoot the ideal value, do this a lot and you dither forever.
- Want learning to converge on right values.

Historical Note

- Our understanding of linear classifiers and probability-based learning came from our attempts to understand what neural networks (NN) could & couldn't do.
- NN are intuitive, easy, algorithmic & attractive, biologically inspired.
- But from about 1990, the real action was happening in straight maths.

Neat vs Scruffy

- How can you be sure your problem is linearly separable?
  - You can't. Just try it. Scruffy.
  - Only use methods in situations you can prove the outcome for. Neat.
- Neat methods once known tend to work well, but may take unnecessarily long or overlook solvable problems.

Neats + Scruffies

- A collection of hacks is more likely to win if it is motivated by theory – if each hack is a reasonable approximation of what a sound system would do.
- Neat "hacks" are safer – but never perfect! All computation uses fallible hardware.
- A systems approach can improve safety by using indicators of fail states for scruffy solutions (e.g. the ballooning coefficients.)

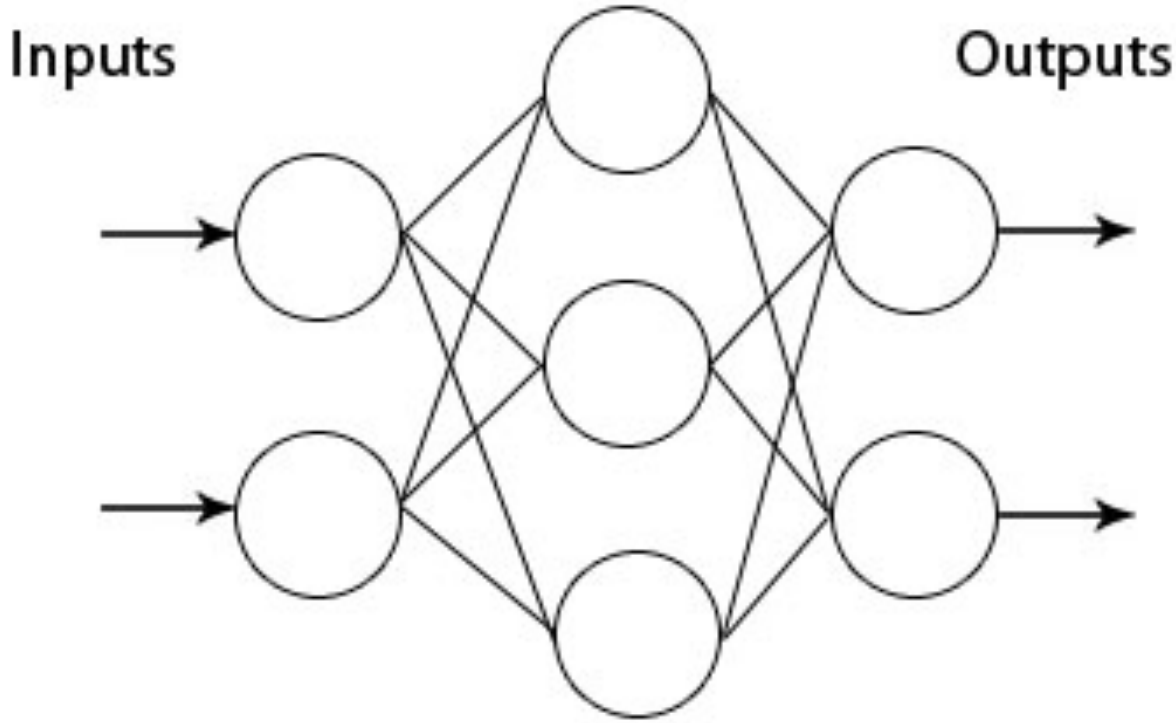
Neats vs Scruffies: Multilayer Perceptrons

- NN "learned like people" will solve AI.
- Minsky & Papert (1969) proved single-layered perceptron networks can't solve some pretty basic problems.
- No one knew how to train multi-layer perceptrons, funding dried up, field almost died.

AI Winter

Multi Layered Perceptron

- Would solve the problem!
- But if there's an error, which weight caused it?

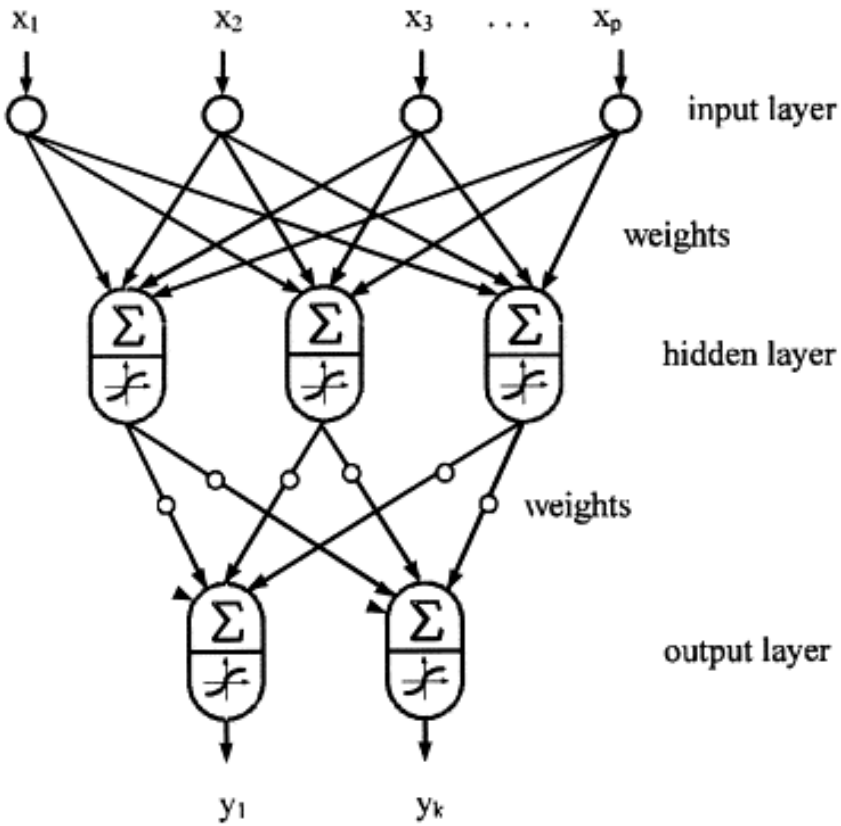


Neats vs Scruffies: Backpropagation

- In the 1980s, several people realised if the threshold was a sigmoid not a step function, you could assign "credit" across layers using calculus – backpropagation.
- But then they realised they could do *lots* of things with calculus & statistics – serious machine learning academics do Bayes now.

(Backpropagation is essentially the chain rule.)

Backpropagation



Geoff Hinton

- One of the (independent) backprop inventors.
- cf. deep learning, Boltzman Machines

Evolution & Genetic Algorithms

Theory & Fact

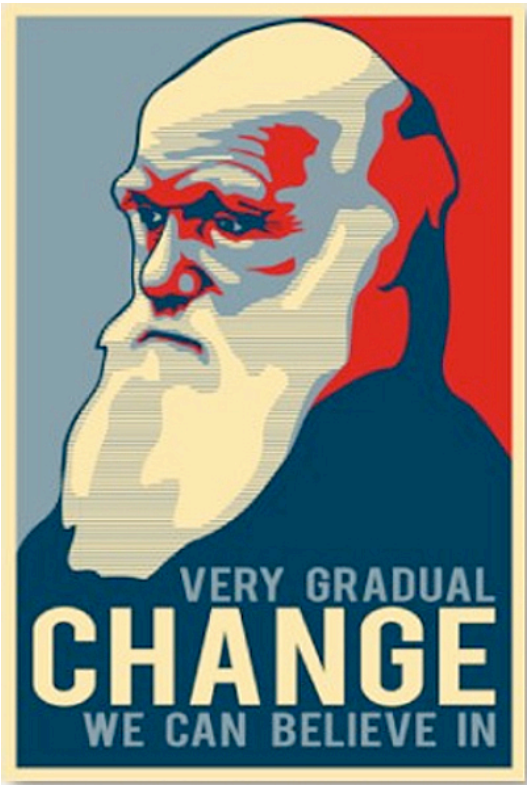
- Evolution: Change over time. Definition
- Evolution of Life: Fact
  - Changes in & diversification of species over time.
- Natural Selection: Theory
  - Current scientific explanation of the observed data.

Fact of Evolution

- Fossil record.
- Observed in laboratory (e.g. with bacteria).
- Genome record of the "tree of life".
- Totally unknown when the theory of evolution (natural selection) was developed.

Darwin's Theory

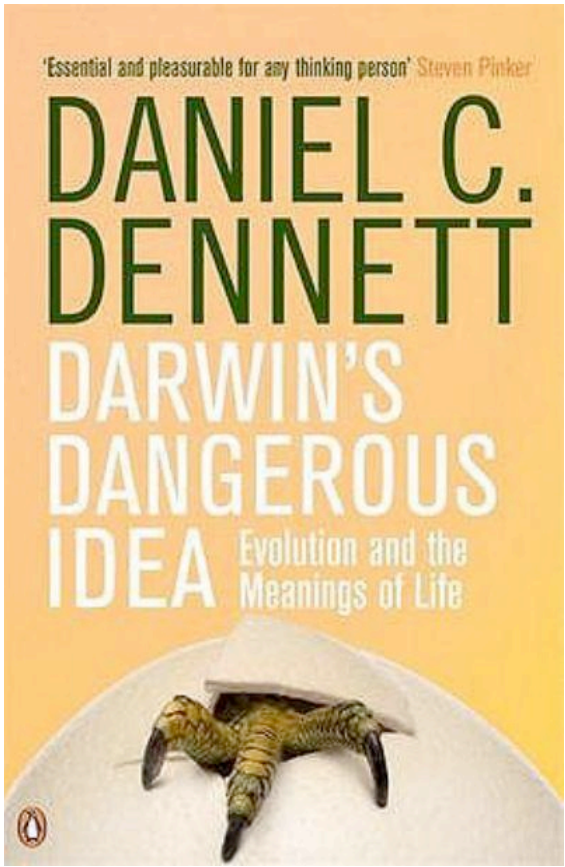
- Animals tend to produce more offspring than survive to reproduce.
- Individuals vary. Offspring more like parents than average for a species.
- The individuals most fit to their environment are most likely to survive and reproduce.



⇒ gradual change

Contemporary Understanding

- Evolution requires variation, reproduction and selection.
- Wherever you have these conditions, you will get change / optimisation to the selection criteria.
- Powerful mechanism for learning / concurrent search.



"Universal acid"

What about the bees?

- Most honey bees have no offspring, but will die for their nest.
- Definition: altruistic behaviour is costly to the individual, but benefits others.
  - Fundamental to sociality, seen even in bacteria.
- How can "survival of the fittest" explain altruistic behaviour?

Replicators

- The mechanisms of heredity (what gets replicated) are called genes.
- Genes are an instruction set that, with the proper biological and environmental context, can produce a new organism.
- Since genes replicate more perfectly than whole animals, and affect behaviour, means social traits like altruism can evolve.

Evolution of Social Traits

- Evolution: variation, selection & transmission.
- What is transmitted is the replicator.
- The unit of selection is the vehicle (or interactor.)
- In the current ecology, most vehicles are composed of many, many replicators.

(Dawkins 1982)

⇒ group selection, kin selection, inclusive fitness

Revision: From Lecture 2

Intelligence & Design

- Combinatorics is the problem, search is the only solution.
- The task of intelligence is to focus search.
  - Called bias (priors) (learning) or constraint (planning).
  - Most `intelligent' behavior has no or little real-time search (non-cognitive) (cf. Brooks IJCAI91).
- For artificial intelligence, most focus from design.

Evolution as Learning

- Bias is provided by phylogeny (evolutionary history), transmitted in the genome.
- Search space is determined by variation in the population.
  - Greater variation accelerates evolution (rate of change, Fisher 1930, Price 1972) but also less exact (remember: "overshooting" with high learning rate).
  - Learning to learn – evolvability.

Science as Evolution

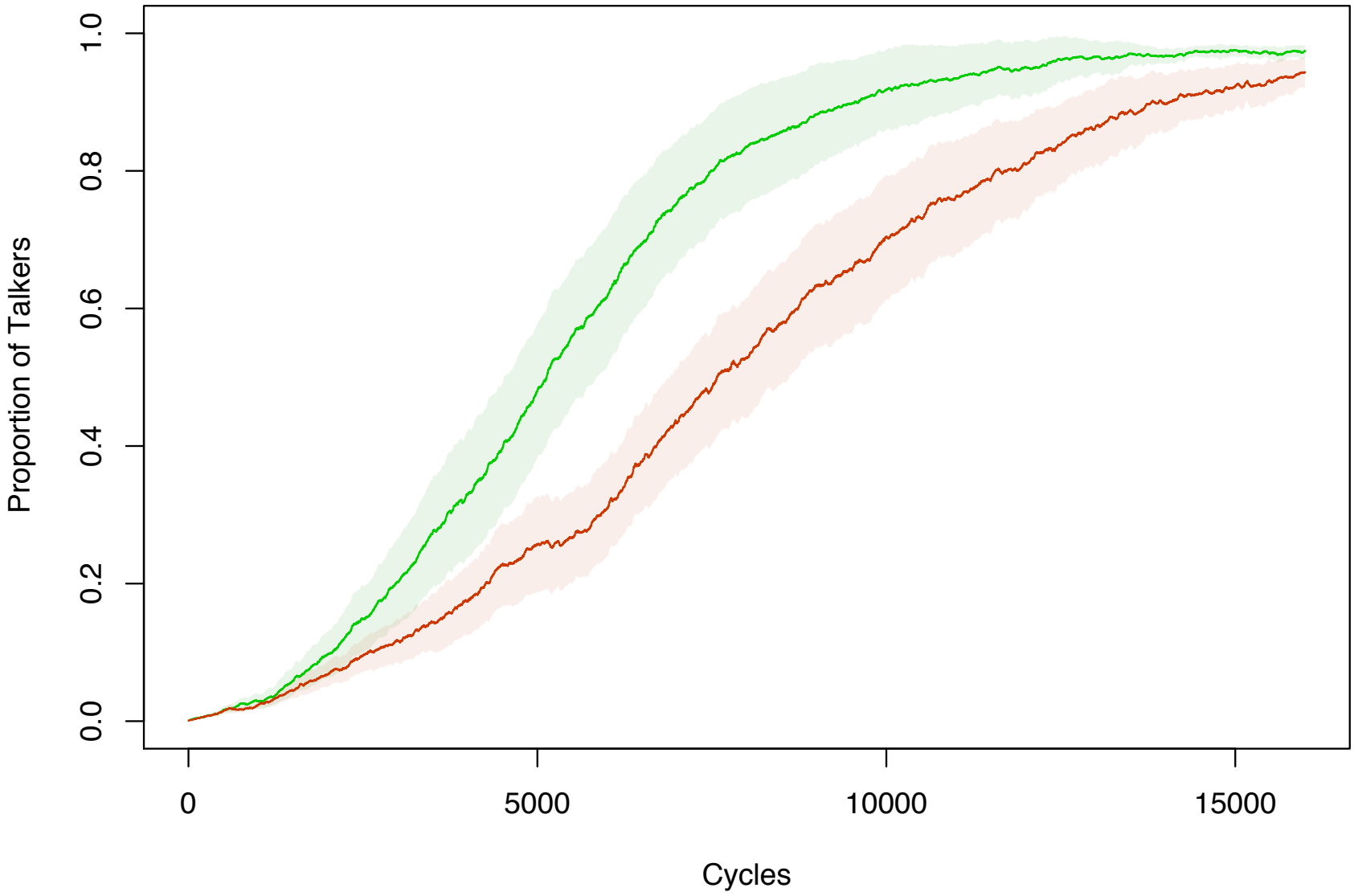
example of memetics–ideas as replicators

- Evolution requires variation, reproduction and selection.
- Variety of theories get taught.
- Theories in new experiments bear some resemblance to what got taught.
- Memory of scientists, peer review, & prediction success perform selection.

Evolution in AI (Genetic Algorithms, GA)

- Variation in some trait.
- Reproduction with inheritance.
  - Often asexual + noise. Sometimes crossover between two or more parents.
- Selection resulting in population change
  - Probability of staying in pool must depend at least partly on differential success.

One trait going to fixation in two different conditions.



Summary

- Machine learning is one way to program AI.
- Requires ways to represent and act on evidence, and to improve evidence based on actions' outcomes.
- Nearest neighbour, neural networks, and genetic algorithms are three (of many!) classes of representations.
- More everywhere on line, in old lecture recordings, and at the end of this file.

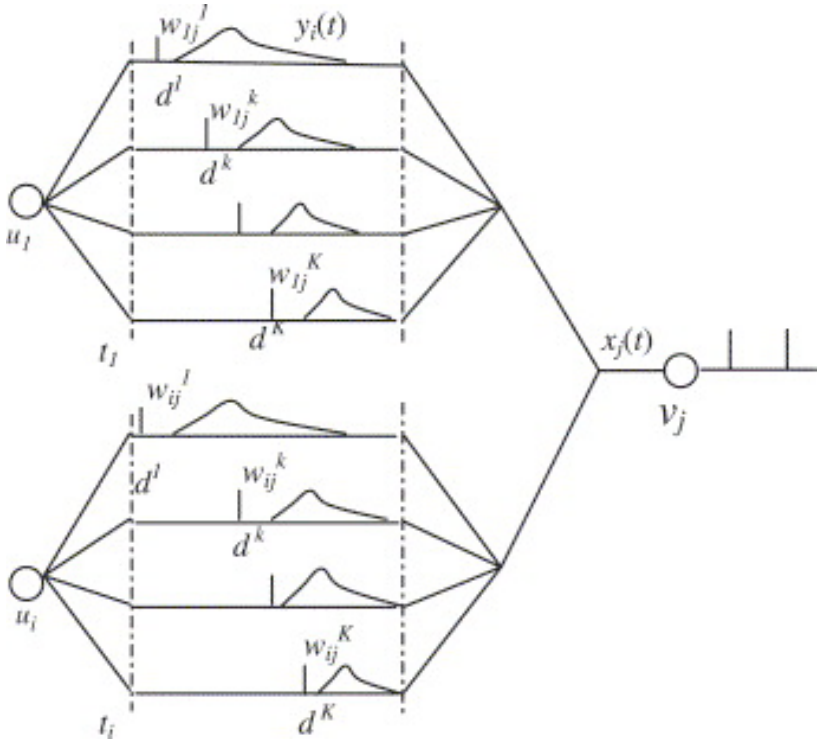
Neats vs Scruffies: Theory vs Practice

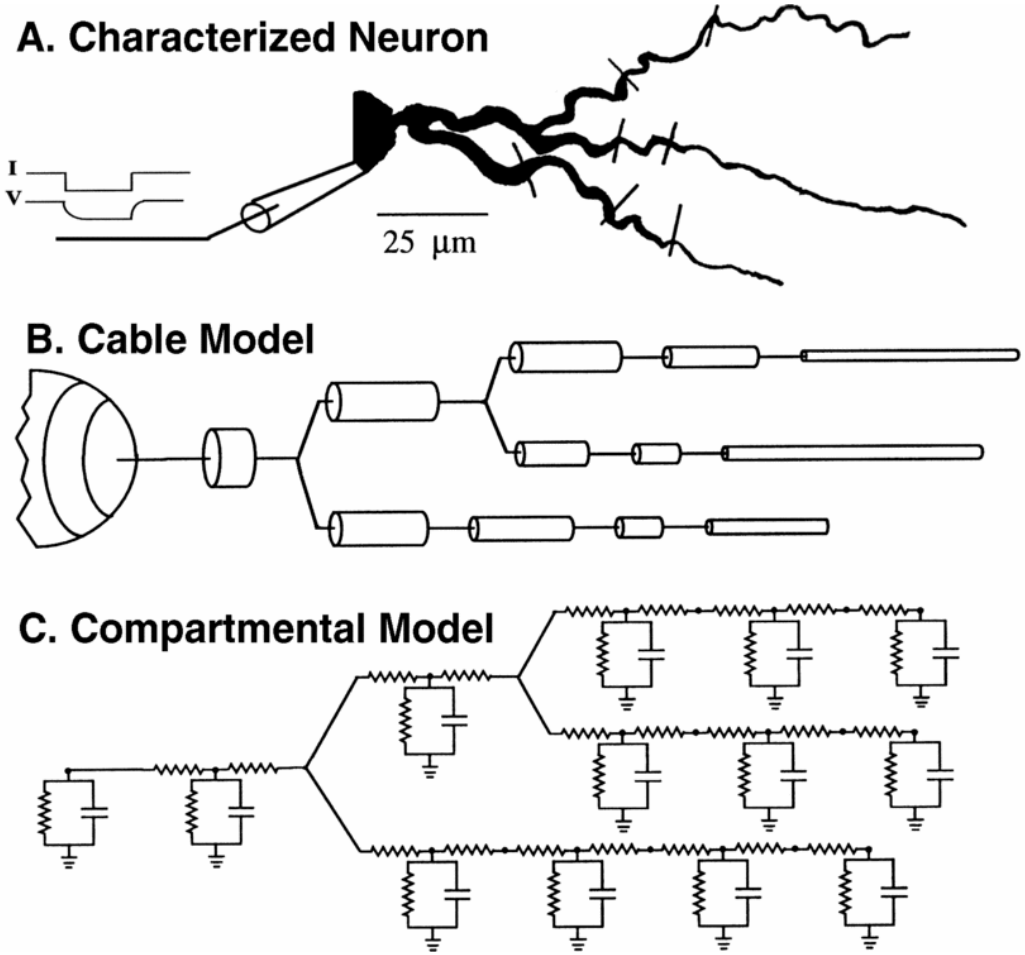
- Serious fast applied stuff e.g. Google do the serious neat stuff (though sometimes scruffily hacked together).
- But many, many, many applications of backpropagation on 3-layer networks in ordinary industry by students like you.
- This bullet in 2013: "NN still used by psychologists, some artificial life researchers."

Also from 2013

Other Topical NN Research

Compartmental models





Spike timing networks

2014 Deep Mind

See also lecture notes…

Mail Online

Home | News | U.S. | Sport | TV&Showbiz | Femail | Health | Science | Money | Video | Coff

Science Home | Pictures | Gadgets Gifts and Toys Store


Surreal seascape revealed by the


Tragedy of woman, 28, who hanged


Rise of the me first mothers: Changing


Ukraine presidency claims breakthrough

British chess prodigy sells artificial intelligence software firm to Google for £242million

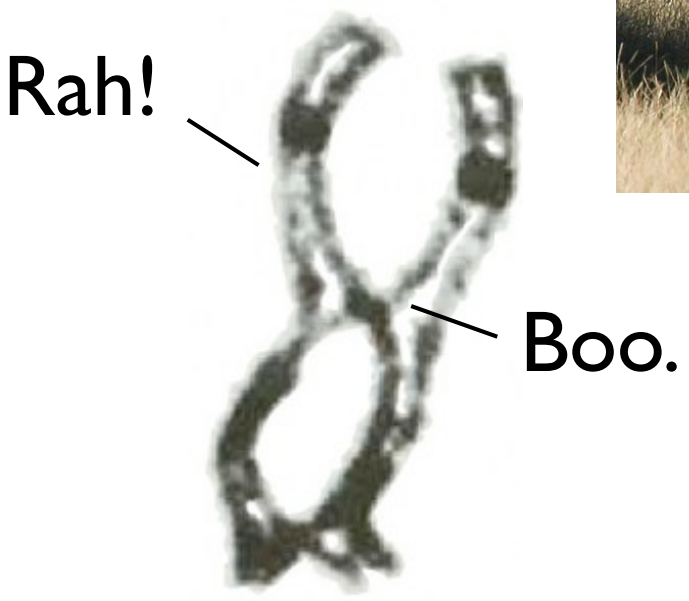
- Neuroscientist Demis Hassabis, 37, co-founded DeepMind two years ago
- London-based firm specialises in 'machine learning'
- The £242million acquisition is Google's biggest-ever in Europe
- Ethics board is said to have been set up to ensure the tech isn't 'abused'
- Facebook was also said to have been in negotiations to buy the firm
- This acquisition follows Google's purchase of seven robotics companies

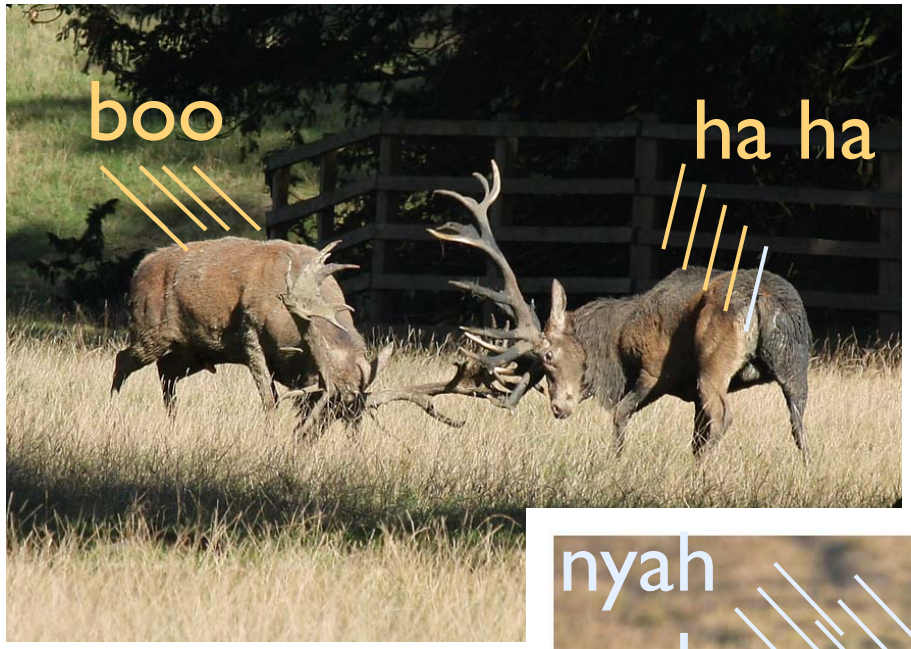
By VICTORIA WOOLLASTON and RUPERT STEINER and AMIE KEELEY

PUBLISHED: 12:05, 27 January 2014 | **UPDATED:** 11:16, 28 January 2014

Multi-Level Selection (different interactors)

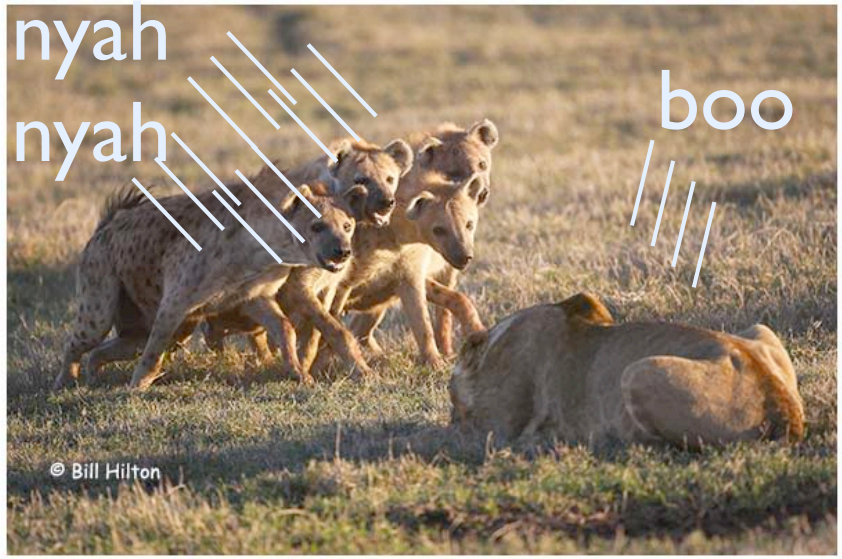
Replicator (Gene)





Organism

Group



What about the bees?

- Most honey bees have no offspring, but will die for their nest.
- All eusocial insects have a 100% monogamous ancestor species ∴ 50% related to sisters.
- In that special case, siblings as useful for propagating genes as offspring.

(Hughes et al 2008)

Questions…

- Are genes the only replicator?
  - Maybe individuals & groups replicate?
  - Maybe memes replicate?
- Evolution is one of the best-supported theories in science, but the details are still constantly being worked out (just like physics.)

Introduction to Genetic Algorithms

(modified) Slides from:

David Hales

www.davidhales.com

Evolution in the real world

- Each cell of a living thing contains ***chromosomes*** - strings of *DNA*
- Each chromosome contains a set of ***genes*** - blocks of DNA
- Each gene determines some aspect of the organism (like eye colour)
- Your set of genes is called a ***genotype***
- Your set of expressed traits is called a ***phenotype***.
- Reproduction involves ***recombination*** of genes from parents and then small amounts of ***mutation*** (errors) in copying
- The ***fitness*** of an organism is how much it can reproduce before it dies (or how much its kids can...)
- Evolution based on "survival of the fittest"

This definition of **gene** is controversial.

The relation between these is very complex.

Dumb AI

A “blind generate and test” algorithm:

Repeat

- Generate a random possible solution
- Test the solution and see how good it is

Until solution is good enough

Can we use this dumb idea?

- Sometimes - yes:
  - if there are only a few possible solutions
  - and you have enough time
  - then such a method *could* be used
- For most problems - no:
  - many possible solutions
  - with no time to try them all
  - so this method *can not* be used

A “less-dumb” idea (GA)

Generate a *set* of random solutions

Repeat

- Test each solution in the set (rank them)
- Remove some bad solutions from set
- Duplicate some good solutions
  - make small changes to some of them

Until best solution is good enough

GA as Evolution

- Evolution requires variation, reproduction and selection.
- Variation from crossover and mutation.
- Reproduce best performers of a set.
- Select best performers based on a fitness function.

GA as Learning

- Learning requires:
  - A representation:
  - A means of acting on current evidence:
  - A means of incorporating feedback:

Representation!

How do you encode a solution?

- Obviously this depends on the problem!
- GA's *often* encode solutions as fixed length "bitstrings" (e.g. 101110, 111111, 000101)
- Each bit represents some aspect of the proposed solution to the problem
- For GA's to work, we need to be able to "test" any string and get a "score" indicating how "good" that solution is

Silly Example - Drilling for Oil

- Imagine you had to drill for oil somewhere along a single 1km desert road
- Problem: choose the best place on the road that produces the most oil per day
- We could represent each solution as a position on the road
- Say, a whole number between [0..1000]

Where to drill for oil?

Solution1 = 300

Solution2 = 900

Road

0

500

1000

Digging for Oil

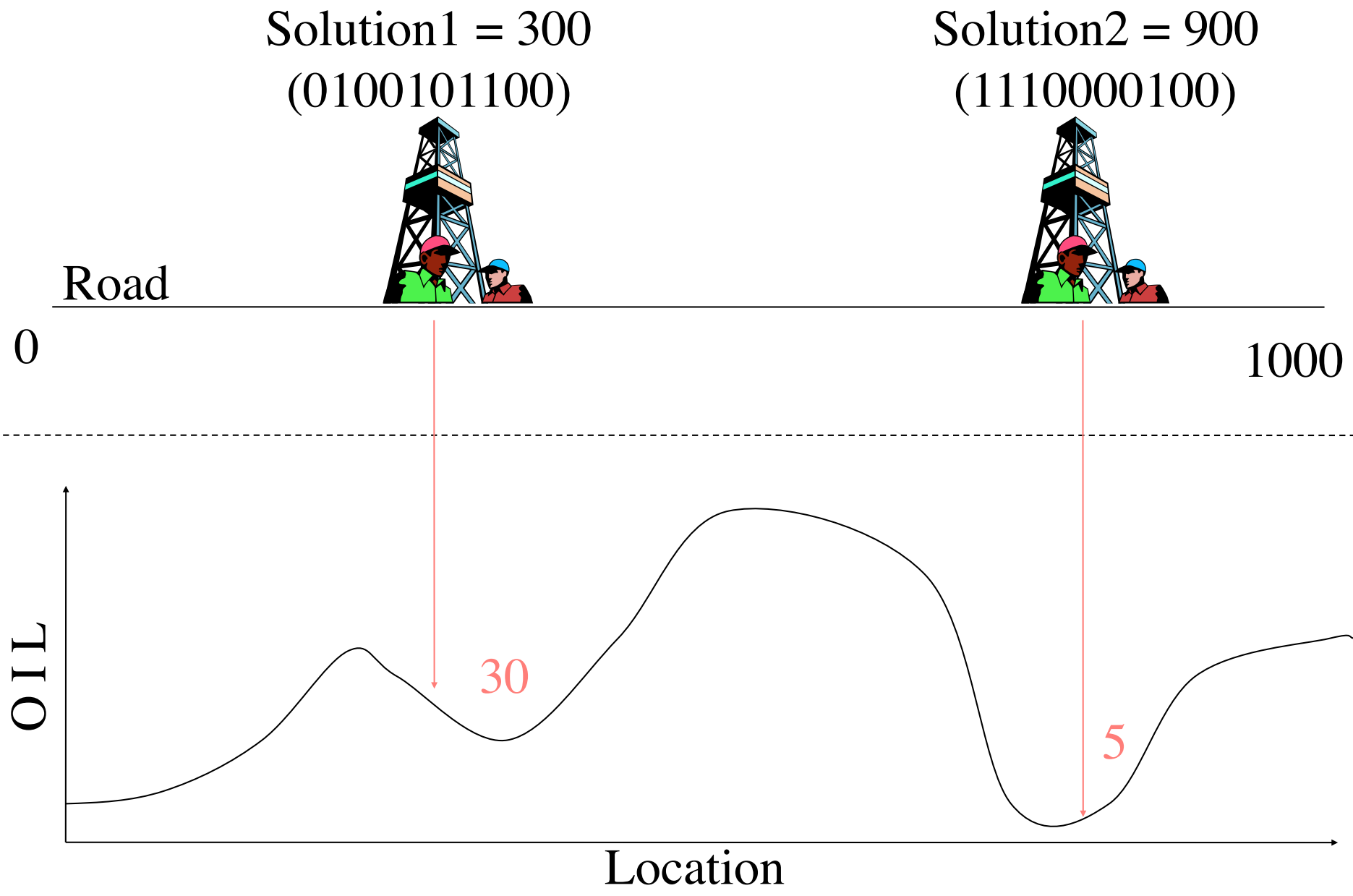
- The set of all possible solutions [0..1000] is called the *search space* or *state space*
  - In this case it's just one number but it could be many numbers or symbols
- Often GA's code numbers in binary producing a bitstring representing a solution
  - In our example we choose 10 bits which is enough to represent 0..1000

Convert to binary string

	512	256	128	64	32	16	8	4	2	1
900	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
300	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**0**
1023	1	1	1	1	1	1	1	1	1	1

In GA's these encoded strings are sometimes called "*genotypes*" or "*chromosomes*" *and the individual bits are sometimes called "genes"*

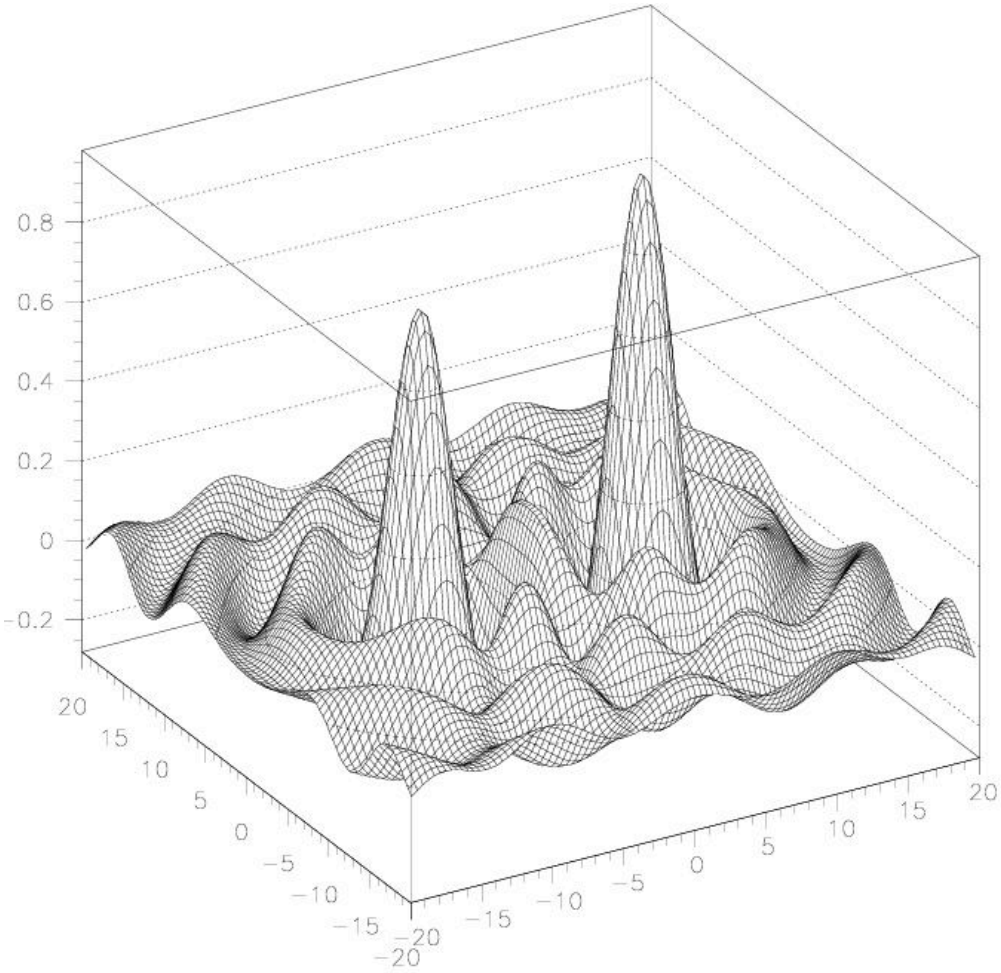
Drilling for Oil

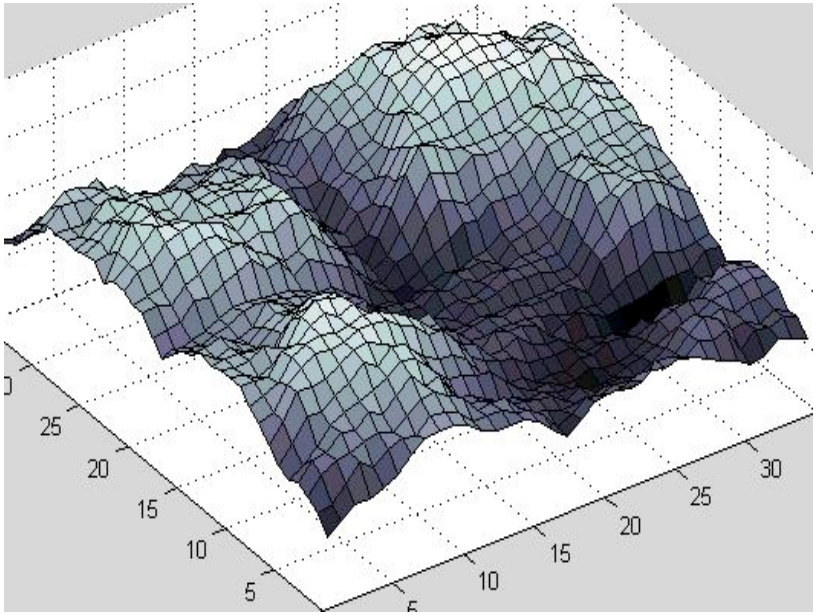


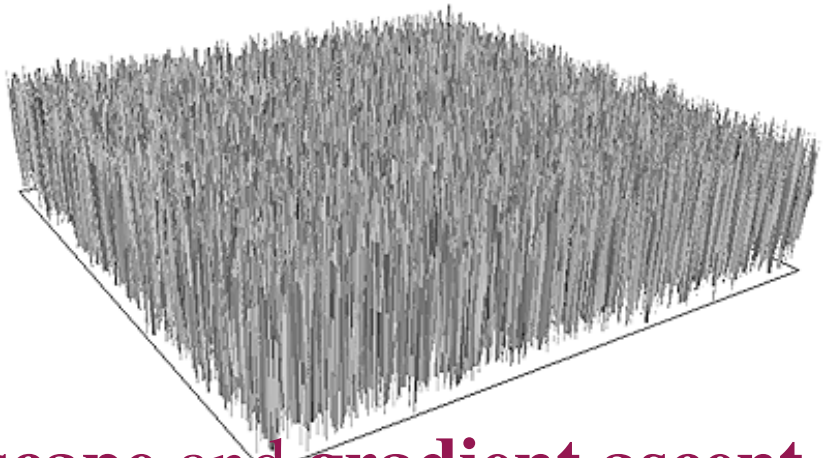
Search Space

- For a simple function f(x) the search space is one dimensional.
- But by encoding several values into the chromosome many dimensions can be searched e.g. two dimensions f(x,y)
- Search space an be visualised as a surface or *fitness landscape* in which fitness dictates height
- Each possible genotype is a point in the space
- A GA tries to move the points to better places (higher fitness) in the the space

Fitness landscapes







The **search space**, **fitness landscape** and **gradient ascent** metaphors work for most if not all kinds of learning.

Search Space

- Obviously, the nature of the search space dictates how a GA will perform
- A completely random space would be bad for a GA
- Also GA's can get stuck in local maxima if search spaces contain lots of these
- Generally, spaces in which small improvements get closer to the global optimum are good

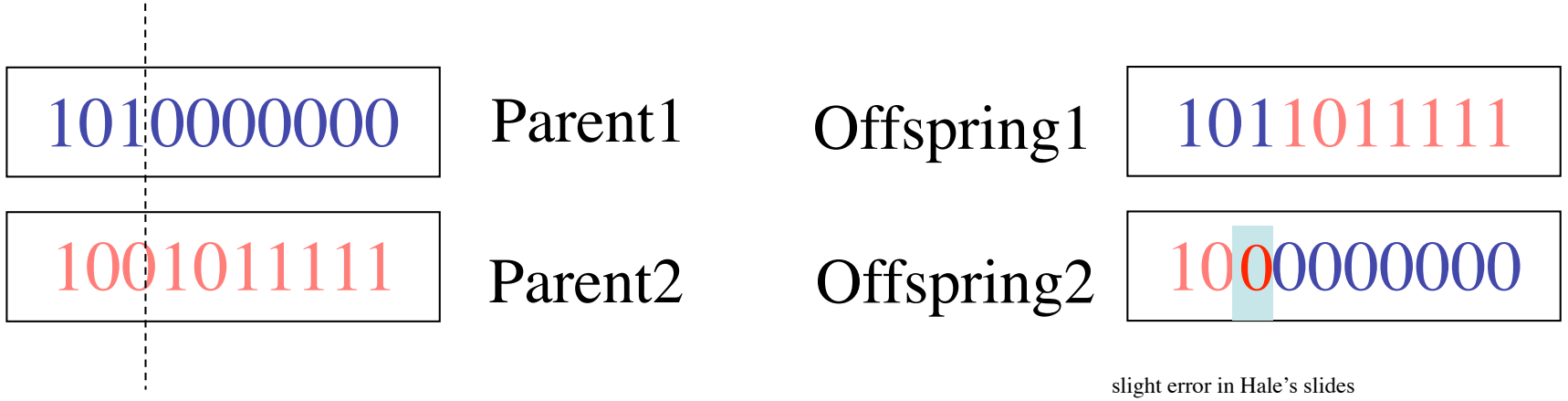
Adding Sex - Crossover

- Although it may work for simple search spaces our algorithm is still very simple
  - It relies on random mutation to find a good solution
- It has been found that by introducing “sex” into the algorithm better results are obtained
- This is done by selecting two parents during reproduction and combining their genes to produce offspring

Adding Sex - Crossover

- Two high scoring “parent” bit strings (*chromosomes)* are selected and with some probability (crossover rate) combined
- Producing two new *offspring* (bit strings)
- Each offspring may then be changed randomly (*mutation*)

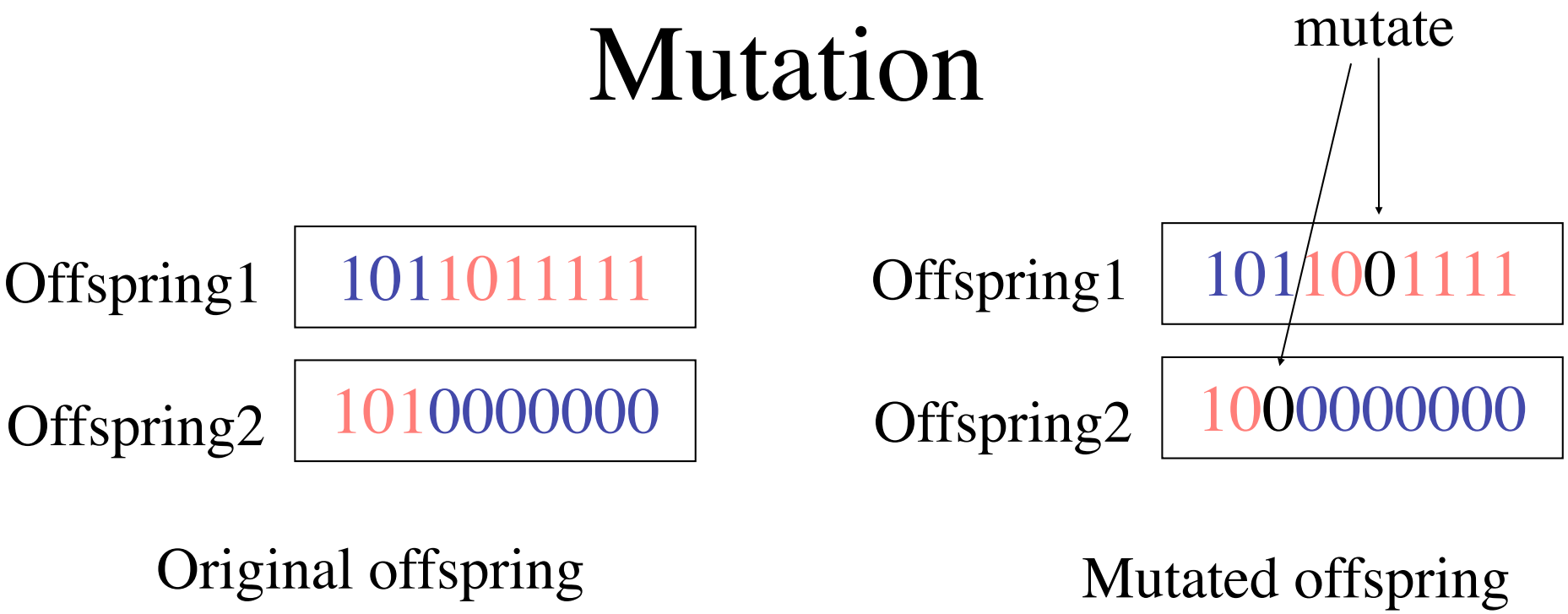
Crossover - Recombination



Crossover single point - random

With some high probability (*crossover rate*) apply crossover to the parents. (*typical values are 0.8 to 0.95*)

Mutation



With some small probability (the *mutation rate*) flip each bit in the offspring (*typical values between 0.1 and 0.001*)

Many Variants of GA

- Different kinds of selection (not roulette)
  - Tournament
  - Elitism, etc.
- Different recombination
  - Multi-point crossover
  - 3 way crossover etc.
- Different kinds of encoding other than bitstring
  - Integer values
  - Ordered set of symbols
- Different kinds of mutation

AI can use many more types than strictly genetic, but if you add in social behaviour (let alone memetics) nature may be using these too.

"3 way etc" = different numbers of parents.

Many parameters to set

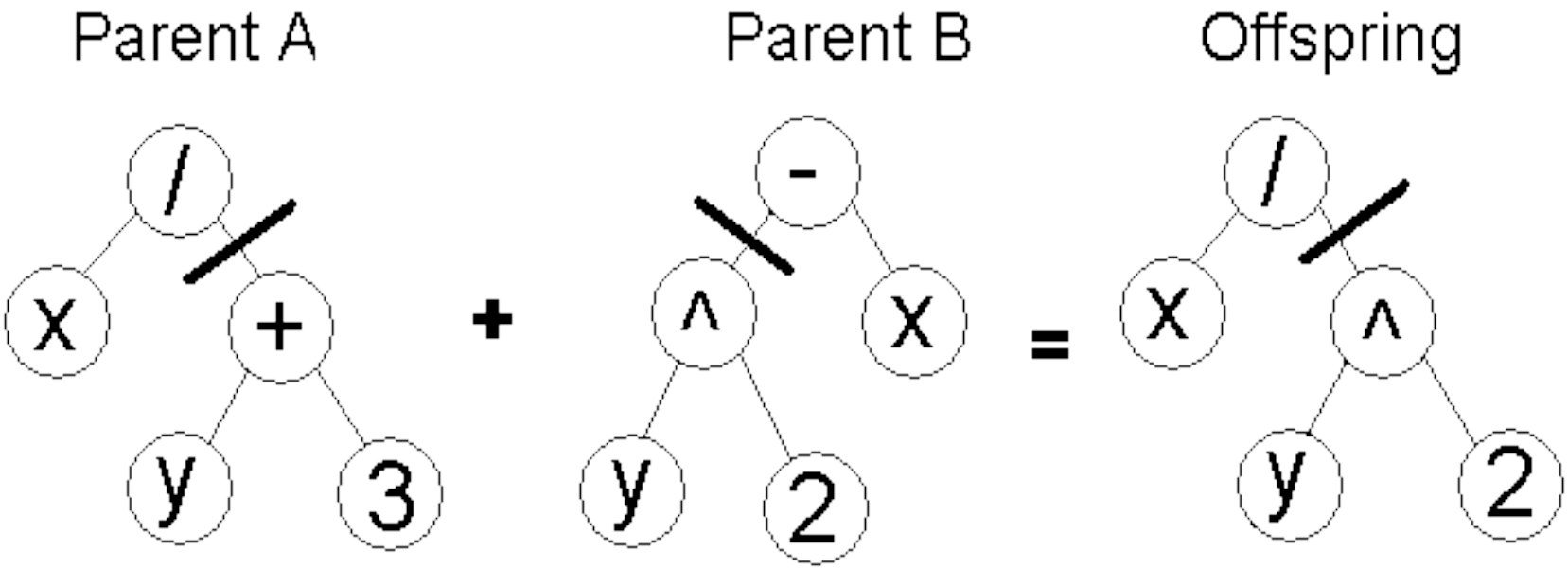
- Any GA implementation needs to decide on a number of parameters: Population size (N), mutation rate (m), crossover rate (c), proportion of agents in next generation, selection function
- Often these are "tuned" based on results obtained (exploration by the programmer) - no general theory to deduce good values

Scruffiness is hard to avoid (DeepMind)

Genetic Programming

- When the chromosome encodes an entire program or function itself this is called genetic programming (GP)
- In order to make this work encoding is often done in the form of a tree representation
- Crossover entails swapping subtrees between parents

Genetic Programming



It is possible to evolve whole programs like this but only small ones. Large programs with complex functions present big problems

History: Genetic Algorithms & Evolutionary Programming

- Pioneered by John Holland in the 1970's
- Got popular in the late 1980's
- GA can be used to solve a variety of problems, but still not well understood.

- Evolutionary or Genetic Programming still unproved.