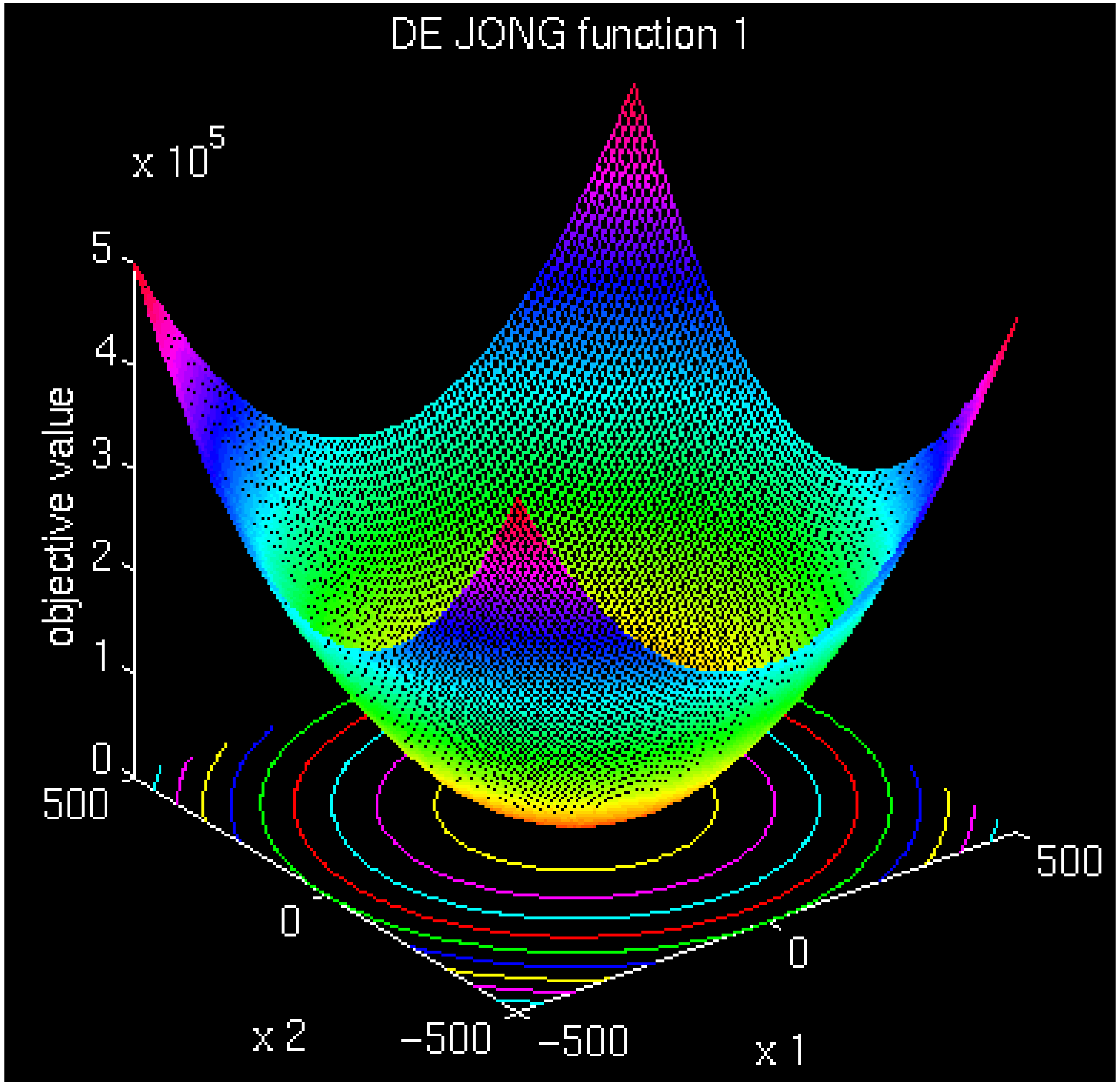
Last words on backprop nets

- Compare Backpropagation & EAs as machine learning methods
- Case Study: combining EAs and backprop for data mining

Remember Fitness Landscapes? (Lecture 3)

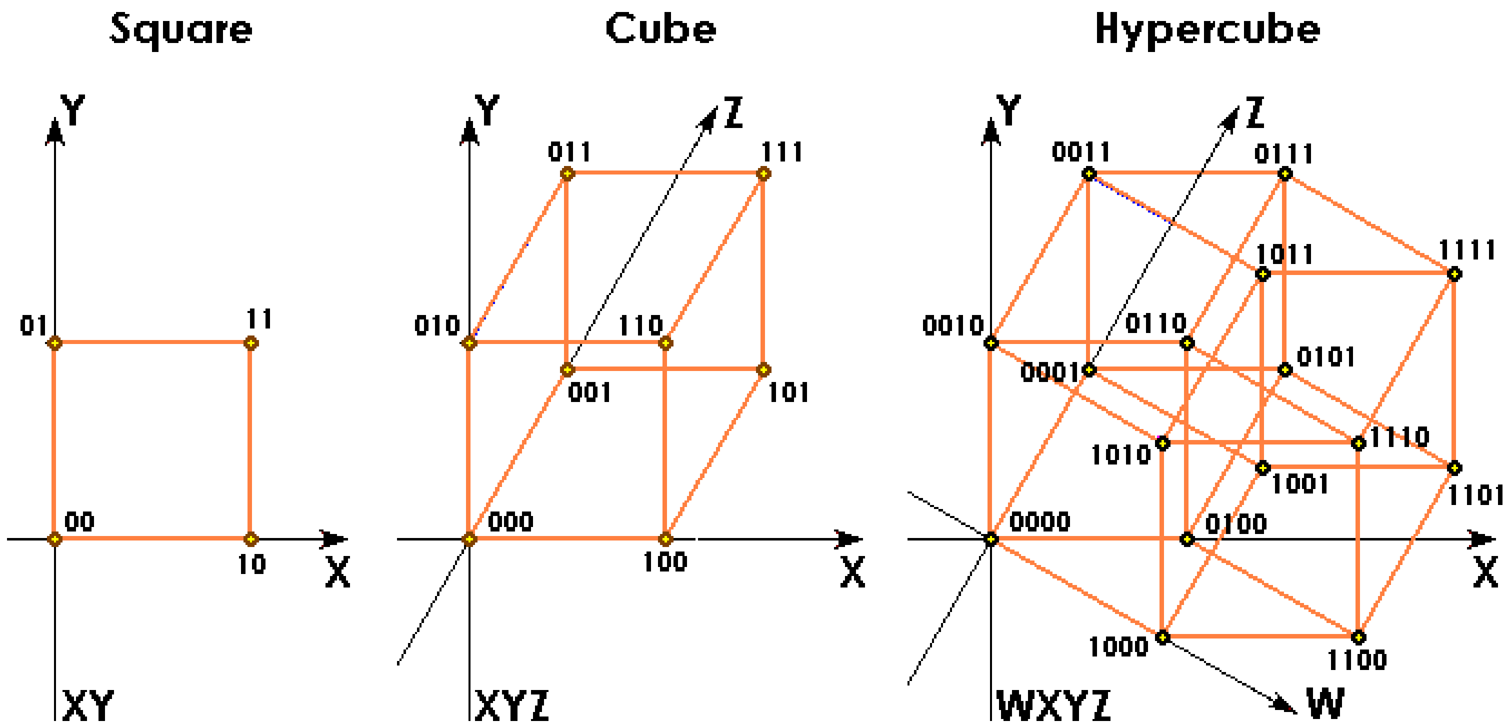
- The notional structure you get by considering (or even plotting) fitness versus genotypes

De Jong Test Function 1: Quadratic Bowl

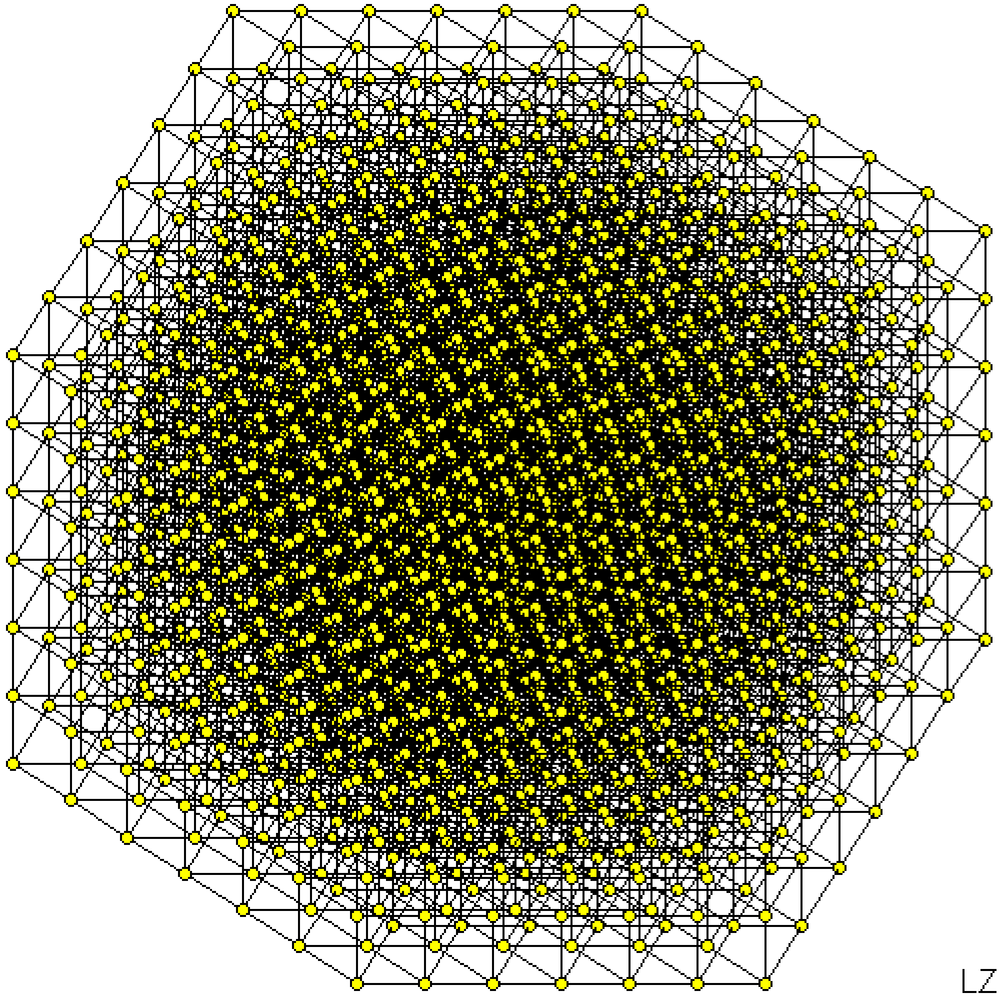


This is what it looks like if there are only two numbers (genes) to optimise

Hypercubes: The topology of the fitness landscape if the genes are binary 0/1 and a mutation just flips a bit



Two Genes Three Genes Four Genes

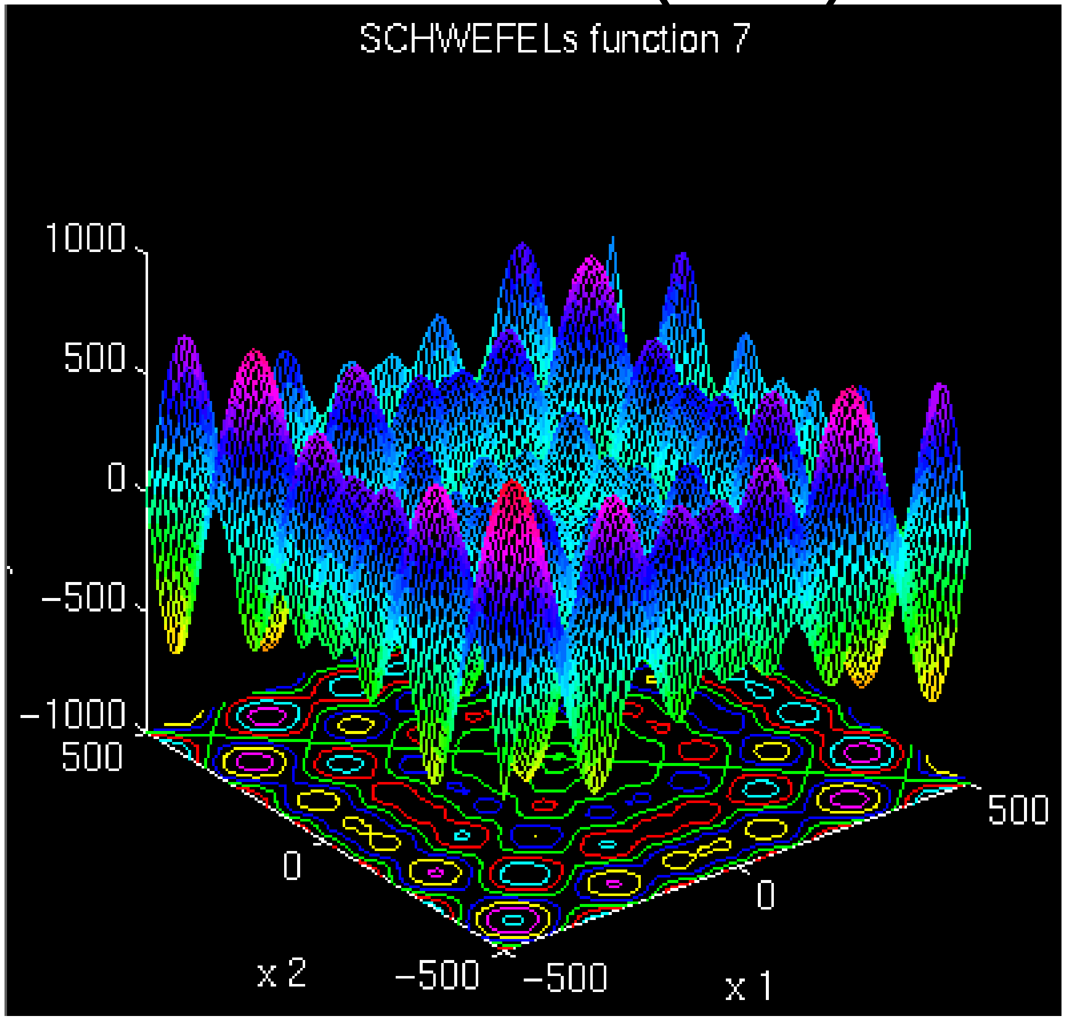


7 genes…

The evolved electronic circuit from Lecture 1 was encoded as 1800 binary genes…

We'll come back to this later when we talk about pathways of neutral mutations.

How Would our (1+1)ES Hill Climber Cope?



f7(x)=sum(-x(i)·sin(sqrt(abs(x(i))))),
i=1:n;
-500<=x(i)<=500.

global minimum
f(x)=-n·418.9829; x(i)=420.9687, i=1:n

-Add an outer loop to repeatedly restart from a new random position every time we get stuck. Such “multiple restart hill climbing” sometimes works very well.

-Use a fancier ES? Lots of work done.

-We’ll look at a standard Genetic Algorithm (GA)

Fitness Landscapes

But the “fitness landscape” (the surface from plotting fitness vs. all possible gene values) is often:

- High-dimensional: difficult/impossible to visualise, can have counterintuitive properties
- Multimodal: contains local optima that can act as “traps”
- Noisy (if based on measurements, complex simulations or stochastic processes)
- Discrete: What if the genes aren’t continuous variables but can only take on a few values?
- Constrained: if some genotypes encode invalid solutions

Fitness Landscape: Advanced

- I simplified before: we'd only seen the (1+1) ES
- Landscape is over an ordering of genotypes, where *neighbours* can easily evolve into each other.
- With small mutation as only variation operator, you can do this but:
  - what happens if you sometimes have many genes mutate at once, or by a large amount? Neighbourhood becomes a stochastic idea.
  - what about crossover?! If a genotype changes by crossing with a very different one, we could jump all over the place: what of neighbourhood then?
- In practice, most mutations small, and individuals become fairly similar quite early in the run – but watch out for lecture on Genetic Programming with vast populations of dissimilar individuals.

Fitness Landscape vs. Error Surface

- We have been talking about perceptrons and multi-layer perceptrons with backpropagation doing “gradient descent” on the error surface.
- Typically define this error as:

$$mean\_squared\_error = \frac{1}{N} \sum_{i \in training/test\_set}^{N} error_i^2$$

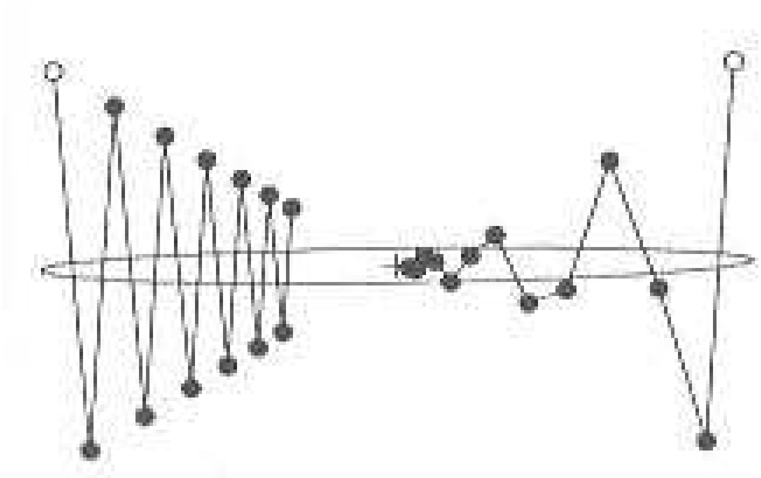


FIGURE 6.3 Gradient descent on the simple quadratic surface of Fig. 5.10. Both trajectories are for 12 steps with $\eta = 0.0476$, the best value in the absence of momentum. On the left there is no momentum ($\alpha = 0$), while $\alpha = 0.5$ on the right.

- The error surface is the error vs. weights structure.
- Learning rules are proven to move you downwards
- This picture from the backprop lectures was for an error surface shaped like the de Jong test function 1 fitness landscape for GAs (quadratic bowl).

Backpropagation vs EA

- Calculates how to adjust weights to move down error surface based on current training example
- Usually averages over small adjustments for each single example, but could also be applied to average error during *batch* presentation

- Makes random changes and keeps the ones that make it no worse
- Fitness trial may consist of various tests to make sure we measure everything we want

Backpropagation vs EA

- Network must be mathematically tractable (to derive the backprop rule)
- Must have detailed training data for supervised learning

- You can try to evolve almost anything without need for mathematical analysis
- Reinforcement learning: fitness value just gives an overall score for performance during the trial – doesn't say exactly what the target should have been at each step

Backpropagation vs EA

- Uses more information: will do better than EA if this is available
- Is a gradient-based method

- Can evolve systems and behaviours that are hard to analyse or specify
- Typically still needs to *generalise* to conditions not seen during evolution
- Is a stochastic search method

Machine Learning

- Backprop and EAs are part of the family of “soft computing” techniques in machine learning. Each is good for different things.

- Next: example of combining backprop with GA for data mining: the GA decides what the inputs to the ANN should be.